Project Title:

GARAGE MANAGEMENT SYSTEM

College Name: **MAILAM ENGINEERING COLLEGE**

Code: **4216**

Department Of Computer Science And Engineering

Team Leader:

Name: Mohamed Salman N	Reg No: 421622104086

Team Members:

Name: Kumaraguru. M	Reg No: 421622104070
Name: Vinothkumar. V	Reg No: 421622104304
Name: Pachaiyappan G	Reg No: 421622104101

Registered Email Id: salmannmohamed55@gmail.com
TrailheadUrl:

Project Name: GARAGE MANAGEMENT SYSTEM

Project overview:

A **Garage Management System (GMS)** is a modern, integrated solution tailored for automotive repair facilities. It is designed to streamline and optimize daily operations, from customer interactions to back-office management. By leveraging technology, GMS enhances service delivery, boosts operational efficiency, and fosters strong customer relationships.

Key Benefits of GMS

- **Efficiency**: Automates routine tasks, such as scheduling and invoicing, allowing staff to focus on delivering quality services.
- **Customer Satisfaction**: Maintains detailed records, enabling personalized communication and timely service reminders.
- **Cost Management**: Tracks inventory and expenses, reducing waste and avoiding stockouts.

- **Data-Driven Decisions**: Provides insightful analytics to identify trends, improve services, and enhance profitability.

Objectives:

The primary aim of a Garage Management System is to optimize the operations of automotive repair facilities while enhancing customer satisfaction and profitability. Below are the key objectives of implementing a GMS:

Business Goals of a Garage Management System (GMS)

The implementation of a **Garage Management System (GMS)** supports the overarching business goals of an automotive repair facility, enabling sustainable growth, improved efficiency, and enhanced customer relationships. Below are the primary business goals aligned with adopting a GMS:

1. Increase Revenue

- Boost sales through better service management and upselling opportunities.
- Retain customers by offering timely reminders for maintenance and promotions.

- Reduce revenue loss by minimizing billing errors and managing overdue payments.

---

2. Optimize Operational Efficiency

- Reduce turnaround time by automating routine tasks and streamlining workflows.
- Minimize waste in labor and resources through effective scheduling and inventory control.
- Improve technician productivity by assigning tasks efficiently.

---

3. Enhance Customer Loyalty

- Build trust with transparent billing and service processes.
- Provide excellent service by maintaining detailed vehicle histories and preferences.
- Deliver a seamless customer experience through digital communication and online booking.

---

4. Control Costs

- Prevent overstocking or understocking with precise inventory management.

- Reduce operational costs by minimizing human errors and inefficiencies.
- Improve profitability by identifying cost-effective services and products.

---

5. Support Business Growth

- Scale operations to manage a larger customer base or multiple locations seamlessly.
- Introduce new services or packages based on customer demand and market trends.
- Stay competitive by adopting cutting-edge technology and industry best practices.

---

6. Foster Data-Driven Decision-Making

- Utilize reports and analytics to identify growth opportunities and performance gaps.
- Make informed decisions about investments, hiring, or marketing strategies.

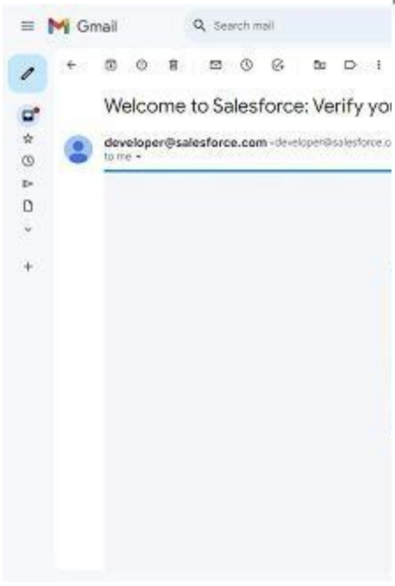
- Stay updated with customer trends to innovate and adapt to changing market demands.

Detailed steps to solution design

Step 1: Creating Developer Account and Activation Creating a developer org in salesforce.

To Activate the account, click on the verify account. Give a password and answer a security question and click on change password.

Give a password and answer a security question and click on change password. Then you will redirect to your salesforce setup page.



Welcome to Salesforce: Verify yo

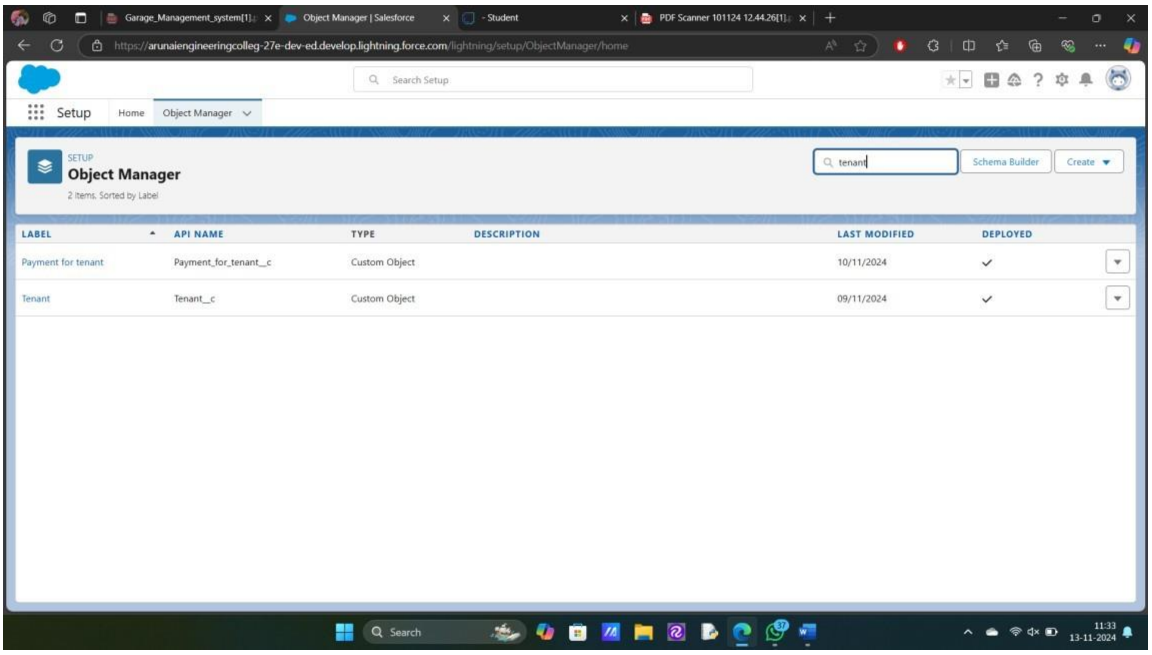
developer@salesforce.com <developer@salesforce.c

to me

Step 2: Create of objects

Create Property Object, tenant object, lease object and payment object.

From the setup page >> Click on Object Manager >>Click on Create >> Click on Custom Object



SETUP
Object Manager
2 Items, Sorted by Label

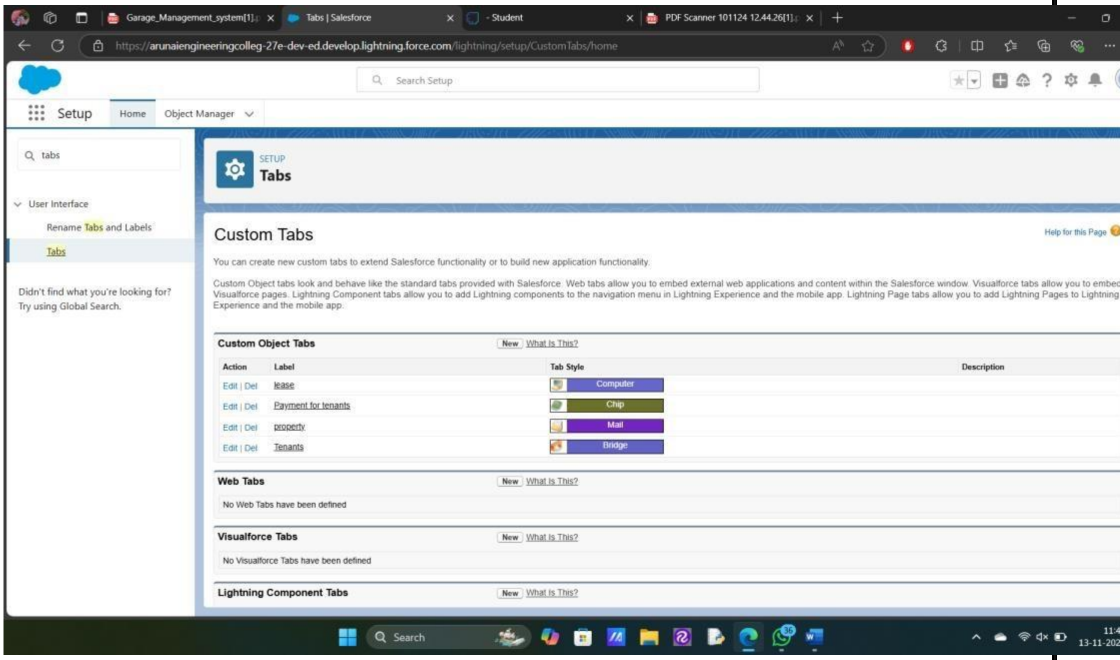
LABEL	API NAME	TYPE	DESCRIPTION	LAST MODIFIED	DEPLOYED
Payment for tenant	Payment_for_tenant__c	Custom Object		10/11/2024	✓
Tenant	Tenant__c	Custom Object		09/11/2024	✓

Enter the label name for the appropriate object. Then enter the Plural label name.

Enter Record Name Label and Format Record Name and Data Type.

Step 3: Creating a Custom Tab

A tab is like a user interface that is used to build records for objects and to view the records in the objects.


Setup | Home | Object Manager

tabs

User Interface
Rename Tabs and Labels
Tabs

Didn't find what you're looking for? Try using Global Search.

SETUP
Tabs

Custom Tabs

You can create new custom tabs to extend Salesforce functionality or to build new application functionality.

Custom Object tabs look and behave like the standard tabs provided with Salesforce. Web tabs allow you to embed external web applications and content within the Salesforce window. Visualforce tabs allow you to embed Visualforce pages. Lightning Component tabs allow you to add Lightning components to the navigation menu in Lightning Experience and the mobile app. Lightning Page tabs allow you to add Lightning Pages to Lightning Experience and the mobile app.

Custom Object Tabs

Action	Label	Tab Style	Description
Edit | Del	lease	Computer	
Edit | Del	Payment for tenants	Chip	
Edit | Del	property	Mail	
Edit | Del	Tenants	Bridge	

Web Tabs

No Web Tabs have been defined

Visualforce Tabs

No Visualforce Tabs have been defined

Lightning Component Tabs

Step 4: Create a Lightning App

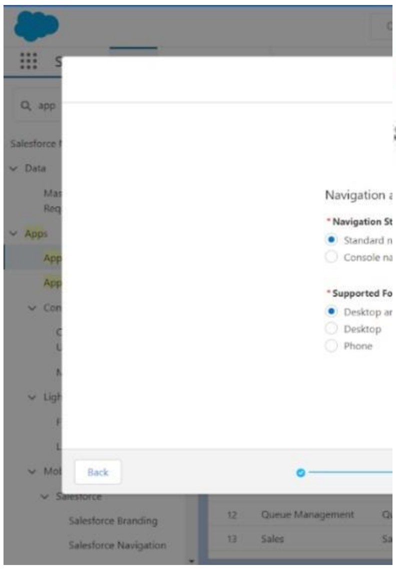
An app is a collection of items that work together to serve a particular function.

To create a Tab:(Property)

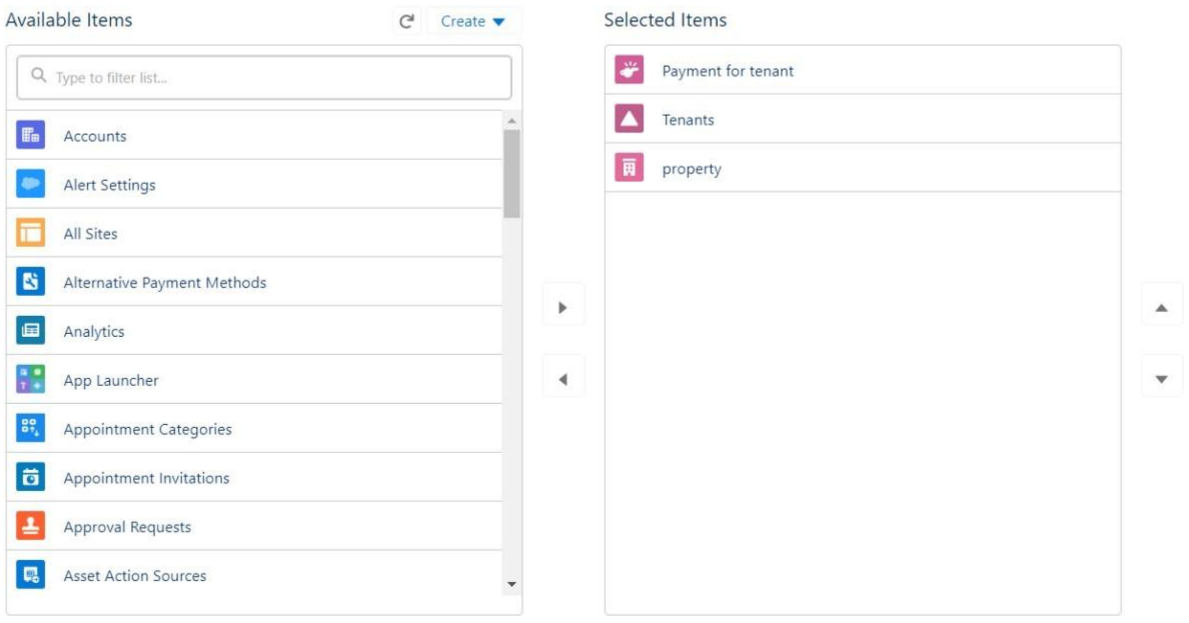
Goto set up page. Search tabs on quick find bar click new under custom tab.

The first step is to select object property and tab style.

Make sure that the Append tab to users' existing personal customizations is checked. Click save



Create a lightning app with an App Name of "garage management system"



Available Items

Create

Type to filter list...

- Accounts
- Alert Settings
- All Sites
- Alternative Payment Methods
- Analytics
- App Launcher
- Appointment Categories
- Appointment Invitations
- Approval Requests
- Asset Action Sources

Selected Items

- Payment for tenant
- Tenants
- property

Step 5: Field Creation

Creation of fields for the customer details , lookup ,checkbox , date ,currency,

Text ,picklist and formula field in service records object.

Go to setup then click on Object Manager to type object name(Customer Details) in search bar and click on the object.

Field Label :Gmail

Field Name : gets auto generated

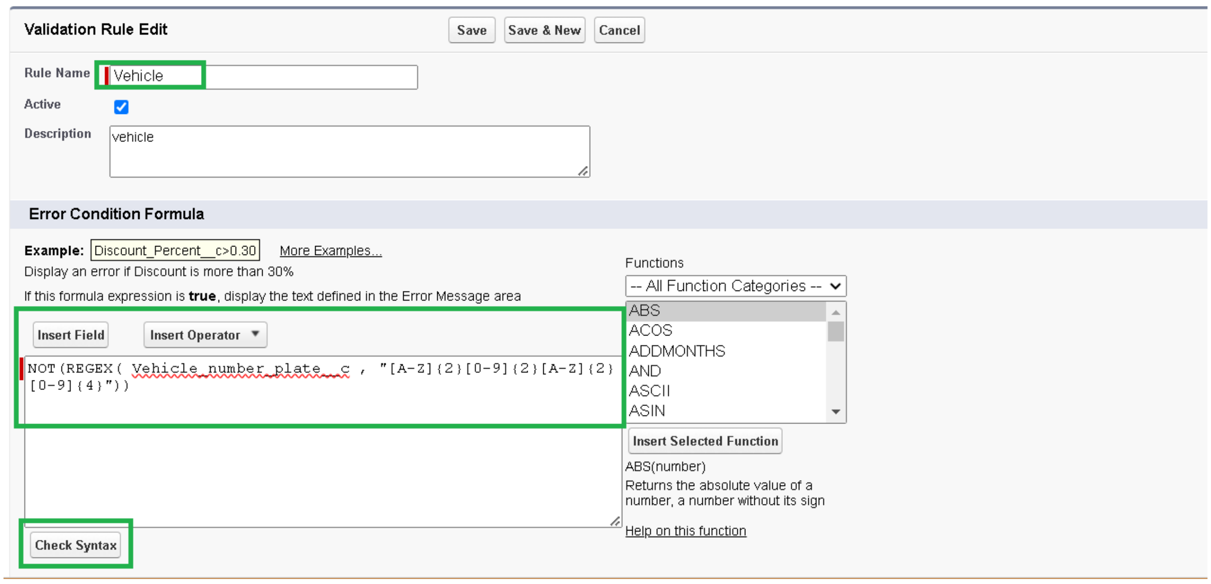
Click on required check box

Click on Next >> Next>> Save and new.

Field Label :Phone number

Field Name : gets auto generated

Click on Next >> Next >> Save and new.


Validation Rule Edit

Rule Name	Vehicle
Active	✓
Description	vehicle

Error Condition Formula

Example: Discount_Percent__c>0.30 More Examples...

Display an error if Discount is more than 30%

If this formula expression is **true**, display the text defined in the Error Message area

```
NOT(REGEX( Vehicle_number_plate__c , "[A-Z]{2}[0-9]{2}[A-Z]{2}
[0-9]{4}"))
```

ABS(number)

Returns the absolute value of a number, a number without its sign

Step 6: Validation rules

To create a validation rule to an Appointment object.

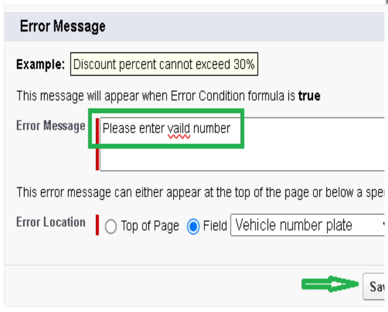
Go to the setup page click on object manager from drop down click edit for Lease object**.**

Click on the validation rule.click new

Rule name= Vehicle

•Insert the Error Condition Formula as :NOT(REGEX(Vehicle_number_plate__c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))

Enter the Error Message as “Please enter vaild number”, select the Error location as Field and select the field as “Vehicle number plate”and click Save.


Error Message

Example: Discount percent cannot exceed 30%

This message will appear when Error Condition formula is **true**

Error Message: Please enter vaild number

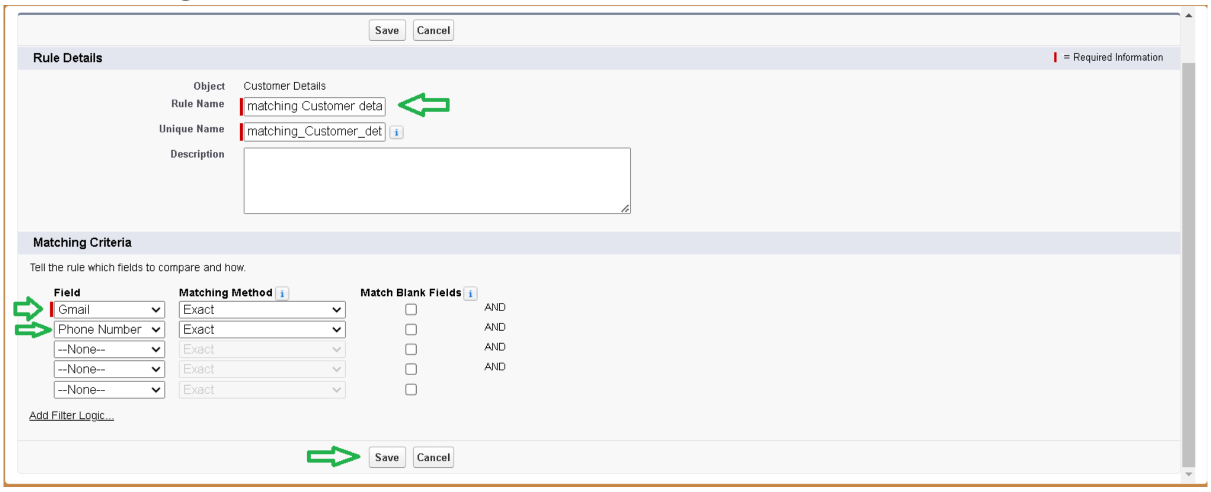
This error message can either appear at the top of the page or below a spe

Error Location: Top of Page [✓ Field] Vehicle number plate

Step 7: Duplicate rule

1.To create a matching rule to an Customer details Object

1. Go to quick find box in setup and search for matching Rule.
2. Click on matching rule >> click on New Rule.
3. Select the object as customer details and click next.
4. Give the Rule name : Matching customer details
5. Unique name : is auto populated
6. Define the matching criteria as
7. Field Matching Method
8. 1. Gmail Exact
9. 2. Phone Number Exact
10. Click save.
11. After Saving Click on Activate.



Rule Details

| = Required Information

Object: Customer Details
Rule Name: matching Customer deta
Unique Name: matching_Customer_det
Description:

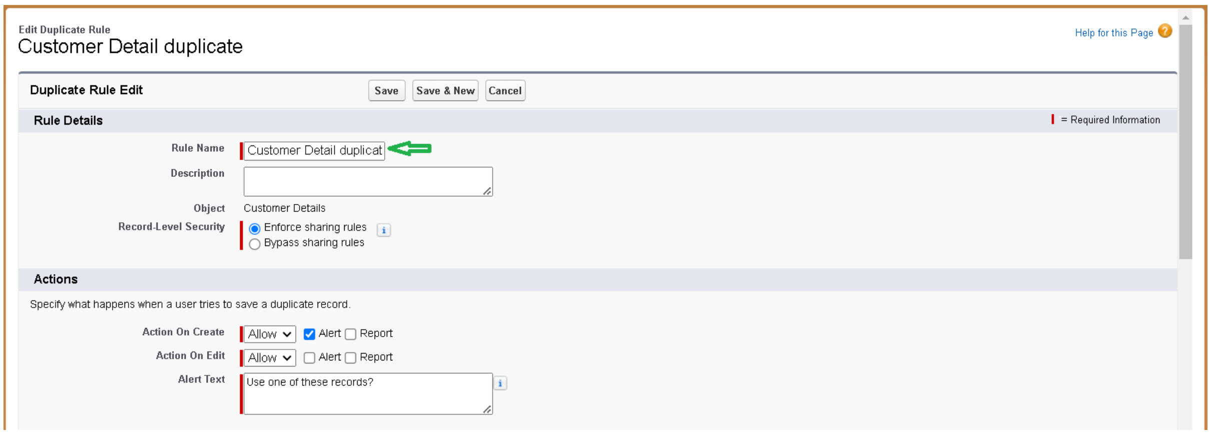
Matching Criteria

Tell the rule which fields to compare and how.

Field	Matching Method	Match Blank Fields	
Gmail	Exact	☐	AND
Phone Number	Exact	☐	AND
--None--	Exact	☐	AND
--None--	Exact	☐	AND
--None--	Exact	☐	

2.To create a Duplicate rule to an Customer details Object

1. Go to quick find box in setup and search for Duplicate rules.
2. Click on Duplicate rule >> click on New Rule >> select customer details object.
3. Give the Rule name as : Customer Detail duplicate
4. Scroll a little in Matching rule section
5. Select the matching rule : Matching customer details
6. And Click on save.
7. After saving the Duplicate Rule, Click on Activate.



Edit Duplicate Rule
Customer Detail duplicate

Duplicate Rule Edit

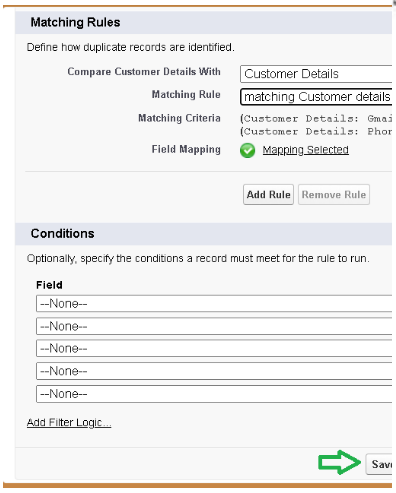
Rule Details — = Required Information

Rule Name	Customer Detail duplicat
Description	
Object	Customer Details
Record-Level Security	[✓ Enforce sharing rules] Bypass sharing rules

Actions

Specify what happens when a user tries to save a duplicate record.

Action On Create	Allow [✓ Alert] Report
Action On Edit	Allow Alert Report
Alert Text	Use one of these records?



Matching Rules

Define how duplicate records are identified.

Compare Customer Details With	Customer Details
Matching Rule	matching Customer details
Matching Criteria	(Customer Details: Gmai (Customer Details: Phon
Field Mapping	Mapping Selected

Conditions

Optionally, specify the conditions a record must meet for the rule to run.

Field
--None--
--None--
--None--
--None--
--None--

Profiles

A profile is a group/collection of settings and permissions that define what a user can do in salesforce. Profile controls “Object permissions, Field permissions, User permissions, Tab settings, App settings, Apex class access, Visualforce page access, Page layouts, Record Types, Login hours & Login IP ranges. You can define profiles by the user's job function. For example System Administrator, Developer, Sales Representative.

Types of profiles in salesforce

1. **Standard profiles:**
   By default salesforce provides below standard profiles.
   - Contract Manager
   - Read Only
   - Marketing User
   - Solutions Manager
   - Standard User
   - System Administrator.

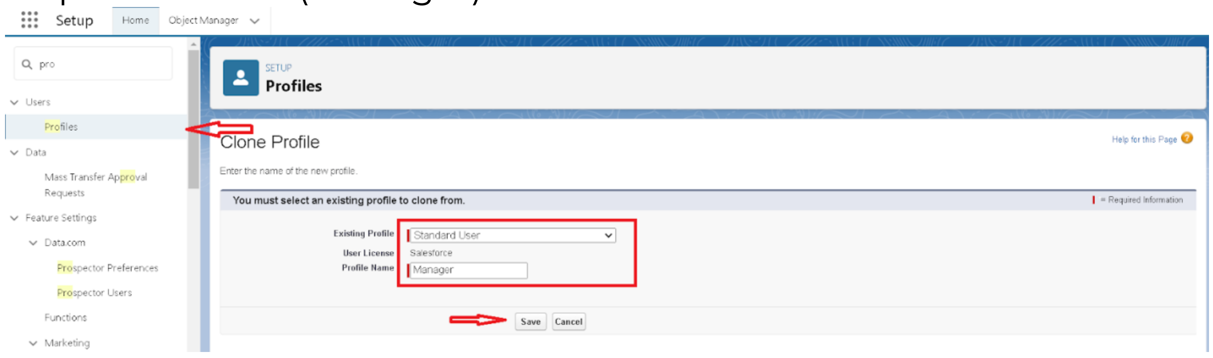
   We cannot deleted standard ones
   Each of these standard ones includes a default set of permissions for all of the standard objects available on the platform.

2. **Custom Profiles:**
   Custom ones defined by us.
   They can be deleted if there are no users assigned with that particular one.

Manager Profile

To create a new profile:

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Manager) >> Save.



Clone Profile

Enter the name of the new profile.

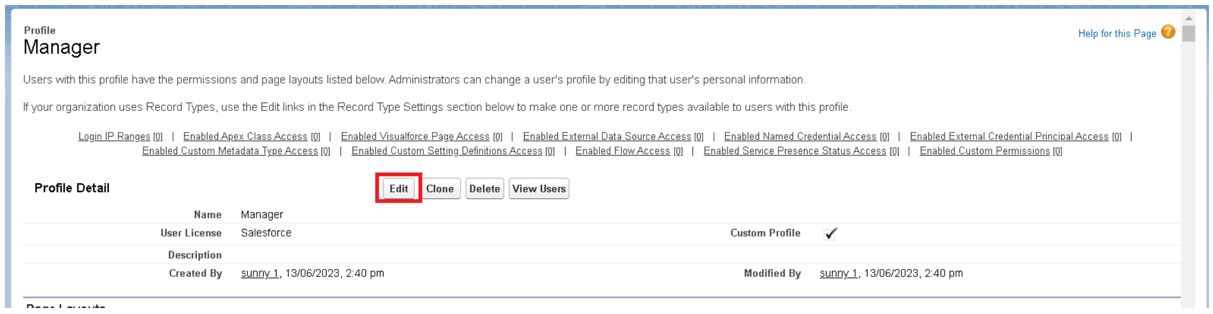
You must select an existing profile to clone from. | = Required Information

Existing Profile: Standard User

User License: Salesforce

Profile Name: Manager

2. While still on the profile page, then click Edit.



Profile

Manager

Users with this profile have the permissions and page layouts listed below. Administrators can change a user's profile by editing that user's personal information.

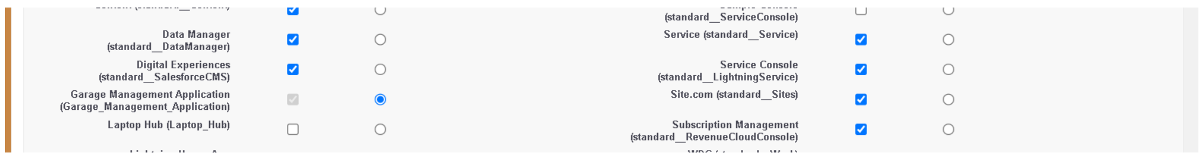
If your organization uses Record Types, use the Edit links in the Record Type Settings section below to make one or more record types available to users with this profile.

Login IP Ranges [0] | Enabled Apex Class Access [0] | Enabled Visualforce Page Access [0] | Enabled External Data Source Access [0] | Enabled Named Credential Access [0] | Enabled External Credential Principal Access [0] | Enabled Custom Metadata Type Access [0] | Enabled Custom Setting Definitions Access [0] | Enabled Flow Access [0] | Enabled Service Presence Status Access [0] | Enabled Custom Permissions [0]

Profile Detail

Name	Manager		
User License	Salesforce	Custom Profile	✓
Description			
Created By	sunny 1, 13/06/2023, 2:40 pm	Modified By	sunny 1, 13/06/2023, 2:40 pm

3. Select the Custom App settings as default for the Garage management.



Data Manager (standard__DataManager)	☑	○	Service (standard__Service)	☑	○
Digital Experiences (standard__SalesforceCMS)	☑	○	Service Console (standard__LightningService)	☑	○
Garage Management Application (Garage_Management_Application)	☑	●	Site.com (standard__Sites)	☑	○
Laptop Hub (Laptop_Hub)	☐	○	Subscription Management (standard__RevenueCloudConsole)	☑	○

4. Scroll down to Custom Object Permissions and Give access permissions for Appointments, Billing details and feedback , service records and customer details objects as mentioned in the below diagram.

5. Changing the session times out after should be " 8 hours of inactivity".
6. Change the password policies as mentioned :
7. User passwords expire in should be " never expires ".
8. Minimum password length should be " 8 ", and click save.

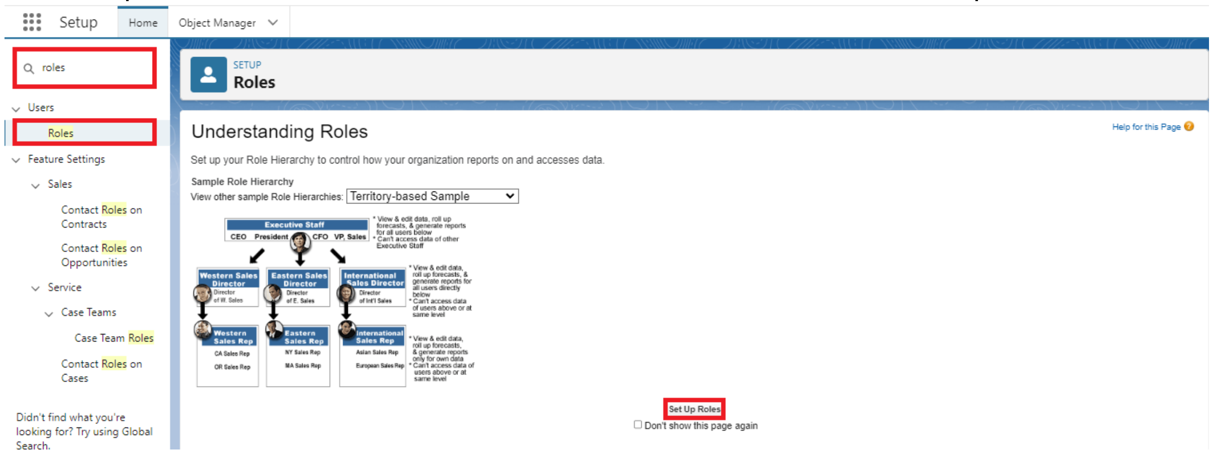
Role & Role Hierarchy

A role in Salesforce defines a user's visibility access at the record level. Roles may be used to specify the types of access that people in your Salesforce organization can have to data. Simply put, it describes what a user could see within the Salesforce organization.

Creating Manager role

Creating Manager Role:

1. Go to quick find >> Search for Roles >> click on set up roles.



Understanding Roles

Set up your Role Hierarchy to control how your organization reports on and accesses data.

Sample Role Hierarchy

View other sample Role Hierarchies: Territory-based Sample

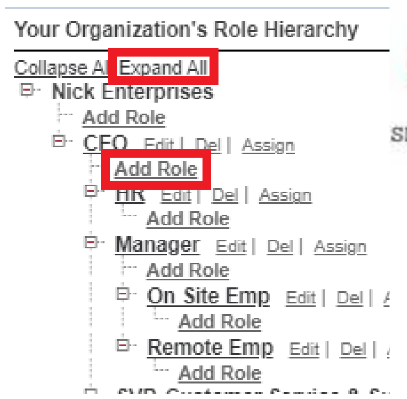
* View & edit data, roll up forecasts, & generate reports for all users below
* Can't access data of other Executive Staff

* View & edit data, roll up forecasts, & generate reports for all users directly below
* Can't access data of users above or at same level

* View & edit data, roll up forecasts, & generate reports only for own data
* Can't access data of users above or at same level

☐ Don't show this page again

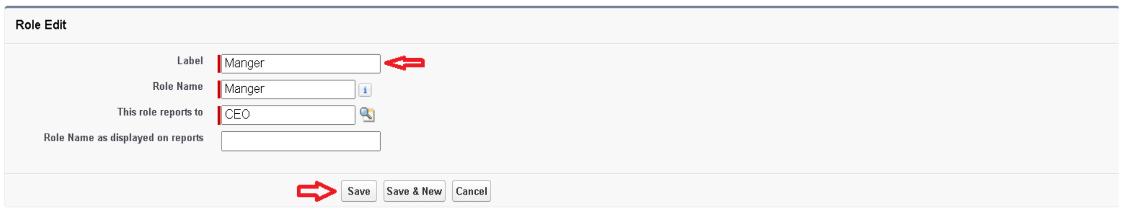
2. Click on Expand All and click on add role under whom this role works.


Your Organization's Role Hierarchy

Collapse All Expand All

Nick Enterprises
- Add Role
- CEO Edit | Del | Assign
  - Add Role
  - HR Edit | Del | Assign
    - Add Role
  - Manager Edit | Del | Assign
    - Add Role
    - On Site Emp Edit | Del |
      - Add Role
    - Remote Emp Edit | Del |
      - Add Role

3. Give Label as “Manager” and Role name gets auto populated. Then click on Save.


Role Edit

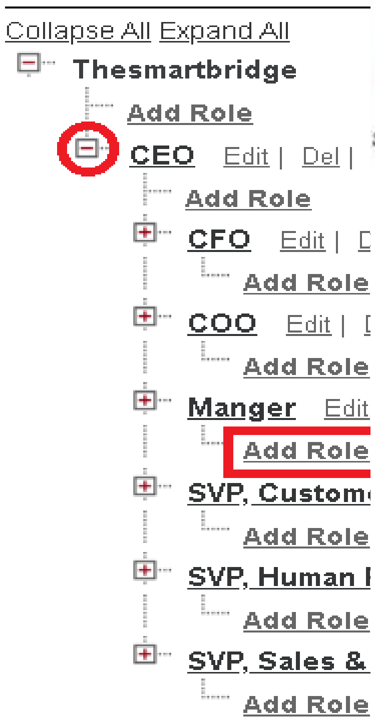
Label	Manger
Role Name	Manger
This role reports to	CEO
Role Name as displayed on reports	

Creating another roles

Creating another two roles under manager

1. Go to quick find >> Search for Roles >> click on set up roles.

2. Click plus on CEO role, and click add role under manager.


Collapse All Expand All
Thesmartbridge
Add Role
CEO Edit | Del |
Add Role
CFO Edit |
Add Role
COO Edit |
Add Role
Manger Edit
Add Role
SVP, Custom
Add Role
SVP, Human
Add Role
SVP, Sales &
Add Role

3. Give Label as “sales person” and Role name gets auto populated. Then click on Save.

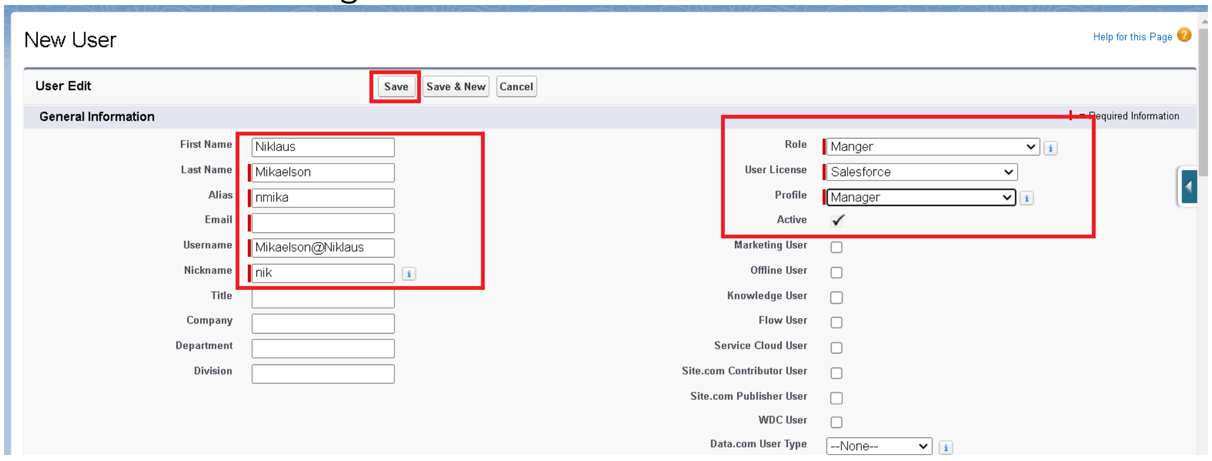
Users

A user is anyone who logs in to Salesforce. Users are employees at your company, such as sales reps, managers, and IT specialists, who need access to

the company's records. Every user in Salesforce has a user account. The user account identifies the user, and the user account settings determine what features and records the user can access.

Create User

1. Go to setup >> type users in quick find box >> select users >> click New user.
2. Fill in the fields
   1. First Name : Niklaus
   2. Last Name : Mikaelson
   3. Alias : Give a Alias Name
   4. Email id : Give your Personal Email id
   5. Username : Username should be in this form: text@text.text
   6. Nick Name : Give a Nickname
   7. Role : Manager
   8. User licence : Salesforce
   9. Profiles : Manager



New User

Help for this Page

User Edit — Save | Save & New | Cancel

General Information — Required Information

First Name	Niklaus	Role	Manger
Last Name	Mikaelson	User License	Salesforce
Alias	nmika	Profile	Manager
Email		Active	✓
Username	Mikaelson@Niklaus	Marketing User	☐
Nickname	nik	Offline User	☐
Title		Knowledge User	☐
Company		Flow User	☐
Department		Service Cloud User	☐
Division		Site.com Contributor User	☐
		Site.com Publisher User	☐
		WDC User	☐
		Data.com User Type	--None--

3. Save.

Creating another users

1. Repeat the steps and create another user using
   a. Role : sales person
   b. User licence : Salesforce Platform

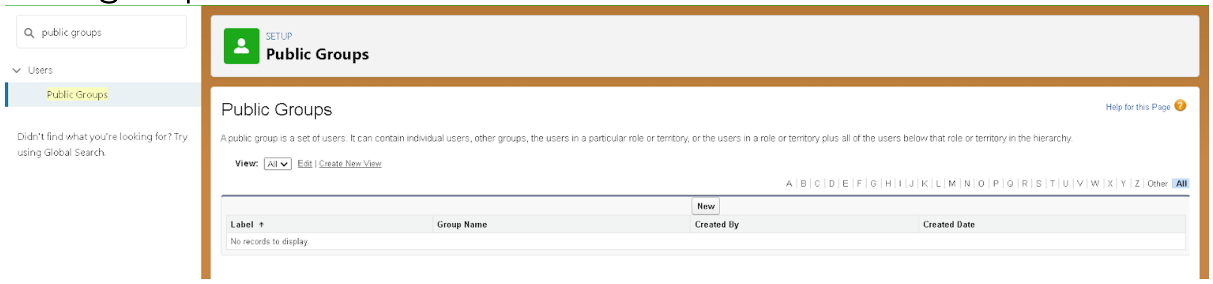
c. Profile : sales person
Note : create atleast 3 users with these permissions.

Public groups

Public groups are a valuable tool for Salesforce administrators and developers to streamline user management, data access, and security settings. By creating and using public groups effectively, you can maintain a secure and organized Salesforce environment while ensuring that users have appropriate access to the resources they need.

Creating new public group

1. Go to setup >> type users in quick find box >> select public groups >> click New.


public groups
Users
Public Groups
SETUP
Didn't find what you're looking for? Try using Global Search.

Public Groups

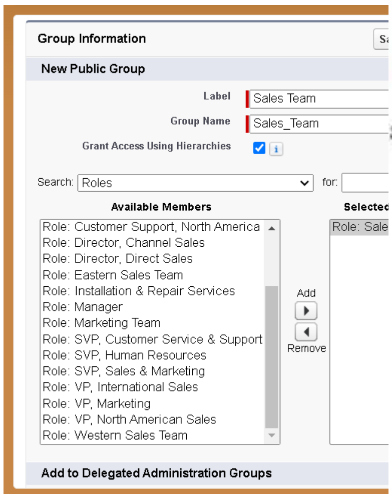
Public Groups

A public group is a set of users. It can contain individual users, other groups, the users in a particular role or territory, or the users in a role or territory plus all of the users below that role or territory in the hierarchy.

Help for this Page
View: All
Edit | Create New View
A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | Other | All
New

Label ↑	Group Name	Created By	Created Date
No records to display.			

2. Give the Label as “sales team”.
3. Group name is autopopulated.
4. Search for Roles.
5. In Available Members select Sales person and click on add it will be moved to selected member.
6. Click on save.


Group Information

New Public Group

Label: Sales Team

Group Name: Sales_Team

Grant Access Using Hierarchies: ✓

Search: Roles for:

Available Members	Selected
Role: Customer Support, North America	Role: Sale
Role: Director, Channel Sales	
Role: Director, Direct Sales	
Role: Eastern Sales Team	
Role: Installation & Repair Services	
Role: Manager	
Role: Marketing Team	
Role: SVP, Customer Service & Support	
Role: SVP, Human Resources	
Role: SVP, Sales & Marketing	
Role: VP, International Sales	
Role: VP, Marketing	
Role: VP, North American Sales	
Role: Western Sales Team	

Add to Delegated Administration Groups

Sharing Setting

Salesforce allows you to configure sharing settings to control how records are accessed and shared within your organization. These settings are crucial for maintaining data security and privacy. Salesforce provides a variety of tools and mechanisms to define and enforce sharing rules, such as:

Organization-Wide Default (OWD) Settings:

These settings define the default level of access for all objects within your Salesforce org.
OWD settings include Private, Public Read-Only, Public Read/Write, and Controlled by Parent.
OWD settings can be configured for each standard and custom object.

Role Hierarchy:

Salesforce uses a role hierarchy to determine record access.

Users at higher levels in the hierarchy have greater access to records owned by or shared with users lower in the hierarchy.
The role hierarchy is often used in combination with OWD settings to grant different levels of access.

Profiles and Permission Sets:

Profiles and permission sets allow administrators to specify object-level and field-level permissions for users.
Profiles are typically used to grant general object and field access, while permission sets can be used to extend those permissions to specific users.

Sharing Rules:

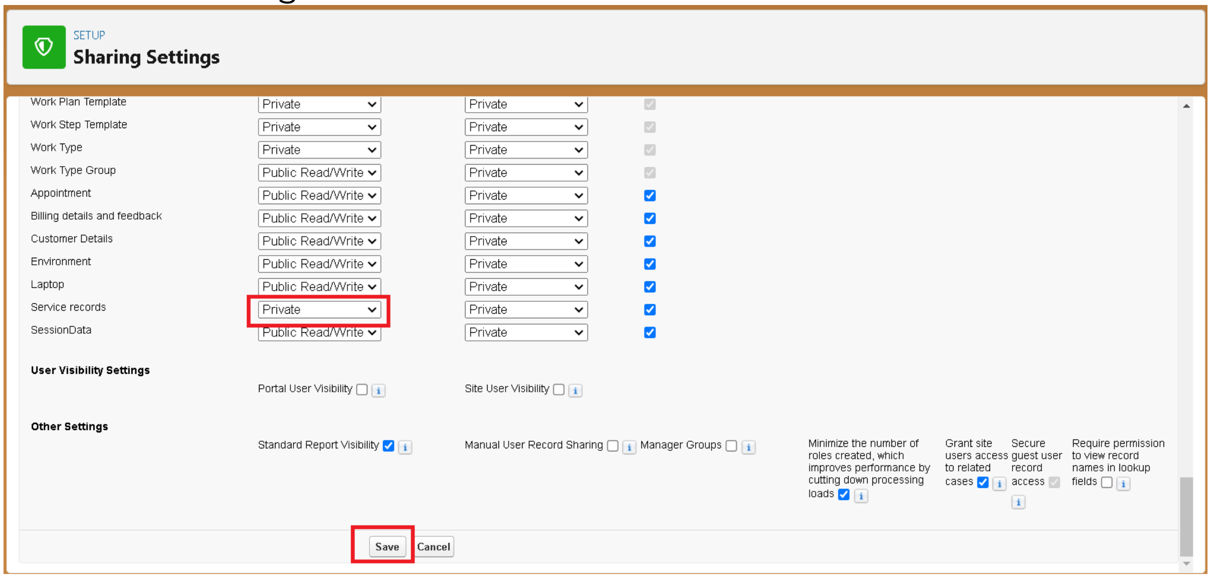
Sharing rules are used to extend access to records for users who meet specific criteria.
They can be used to grant read-only or read-write access to records owned by other users.

Manual Sharing:

Administrators and record owners can manually share specific records with other users or groups.

Creating sharing settings

1. Go to setup >> type users in quick find box >> select Sharing Settings >> click Edit.
2. Change the OWD setting of the Service records Object to private as shown in fig.



SETUP
Sharing Settings

Work Plan Template	Private	Private	✓
Work Step Template	Private	Private	✓
Work Type	Private	Private	✓
Work Type Group	Public Read/Write	Private	✓
Appointment	Public Read/Write	Private	✓
Billing details and feedback	Public Read/Write	Private	✓
Customer Details	Public Read/Write	Private	✓
Environment	Public Read/Write	Private	✓
Laptop	Public Read/Write	Private	✓
Service records	Private	Private	✓
SessionData	Public Read/Write	Private	✓

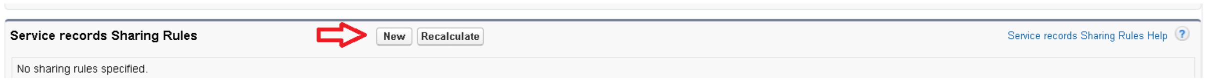
User Visibility Settings

Portal User Visibility ☐ Site User Visibility ☐

Other Settings

Standard Report Visibility ✓ Manual User Record Sharing ☐ Manager Groups ☐ Minimize the number of roles created, which improves performance by cutting down processing loads ✓ Grant site users access to related cases ✓ Secure guest user record access ✓ Require permission to view record names in lookup fields ☐

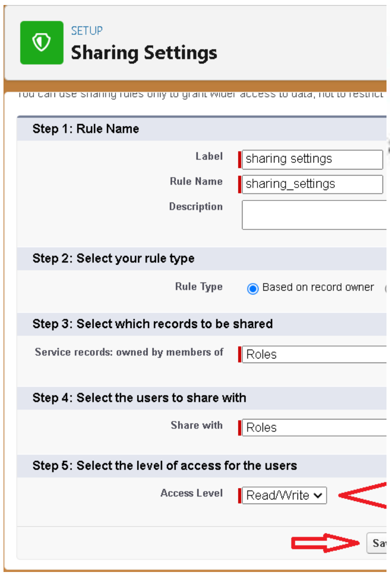
3. Click on save and refresh.
4. Scroll down a bit, Click new on Service records sharing Rules.
5.



Service records Sharing Rules

No sharing rules specified.

6. Give the Label name as " Sharing setting"
7. Rule name is auto populated.
8. In step 3 : Select which records to be shared, members of " Roles " >> " Sales person"
9. In step 4: share with, select " Roles " >> " Manager "
10. In step 5 : Change the access level to " Read / write ".
11. Click on save.


SETUP

Sharing Settings

You can use sharing rules only to grant wider access to data, not to restrict

Step 1: Rule Name

Label	sharing settings
Rule Name	sharing_settings
Description	

Step 2: Select your rule type

Rule Type	● Based on record owner

Step 3: Select which records to be shared

Service records: owned by members of	Roles

Step 4: Select the users to share with

Share with	Roles

Step 5: Select the level of access for the users

Access Level	Read/Write

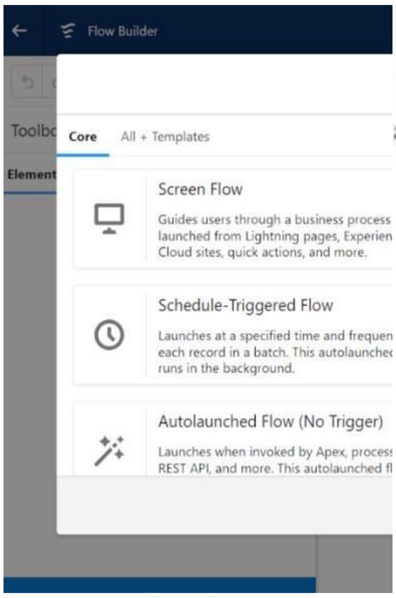
Step 8: Flows

In Salesforce, a flow is a tool that automates complex business processes. Simply put, it collects data and then does something with that data. Flow Builder is the declarative interface used to build individual flows.

Create Flow for monthly payment

Go to setup then type Flow in quick find box Click on the Flow and Select the New Flow.

Select the record Triggered flow. Click on create.



Screen Flow

Guides users through a business process launched from Lightning pages, Experien Cloud sites, quick actions, and more.

Schedule-Triggered Flow

Launches at a specified time and frequen each record in a batch. This autolaunche runs in the background.

Autolaunched Flow (No Trigger)

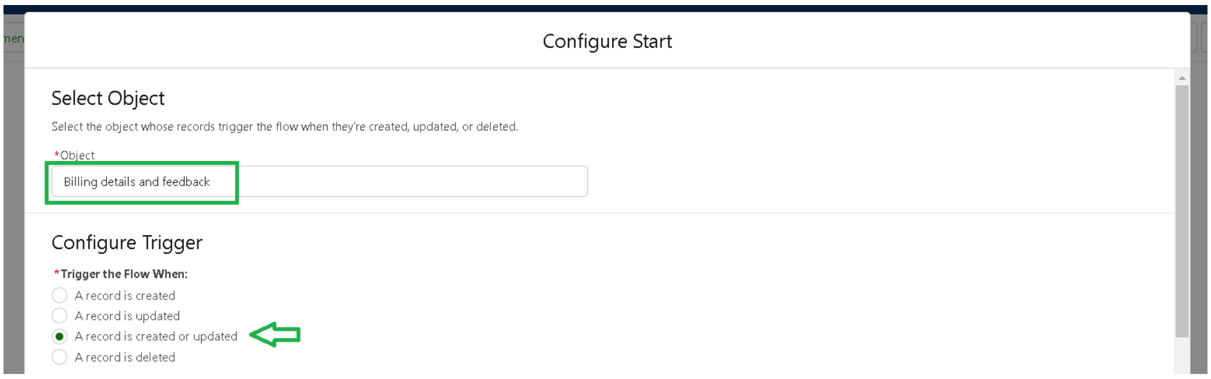
Launches when invoked by Apex, process REST API, and more. This autolaunched fl

Click on: Every time a record is updated and meets the condition requirements Click on: Actions and related records is done

Select the Object as “Billing details and feedback”in the Drop down list.

Select the Trigger Flow when: “A record is Created or Updated”.

Select the Optimize the flow for: “Actions and Related Records” and Click on Done.



Configure Start

Select Object

Select the object whose records trigger the flow when they're created, updated, or deleted.

*Object: Billing details and feedback

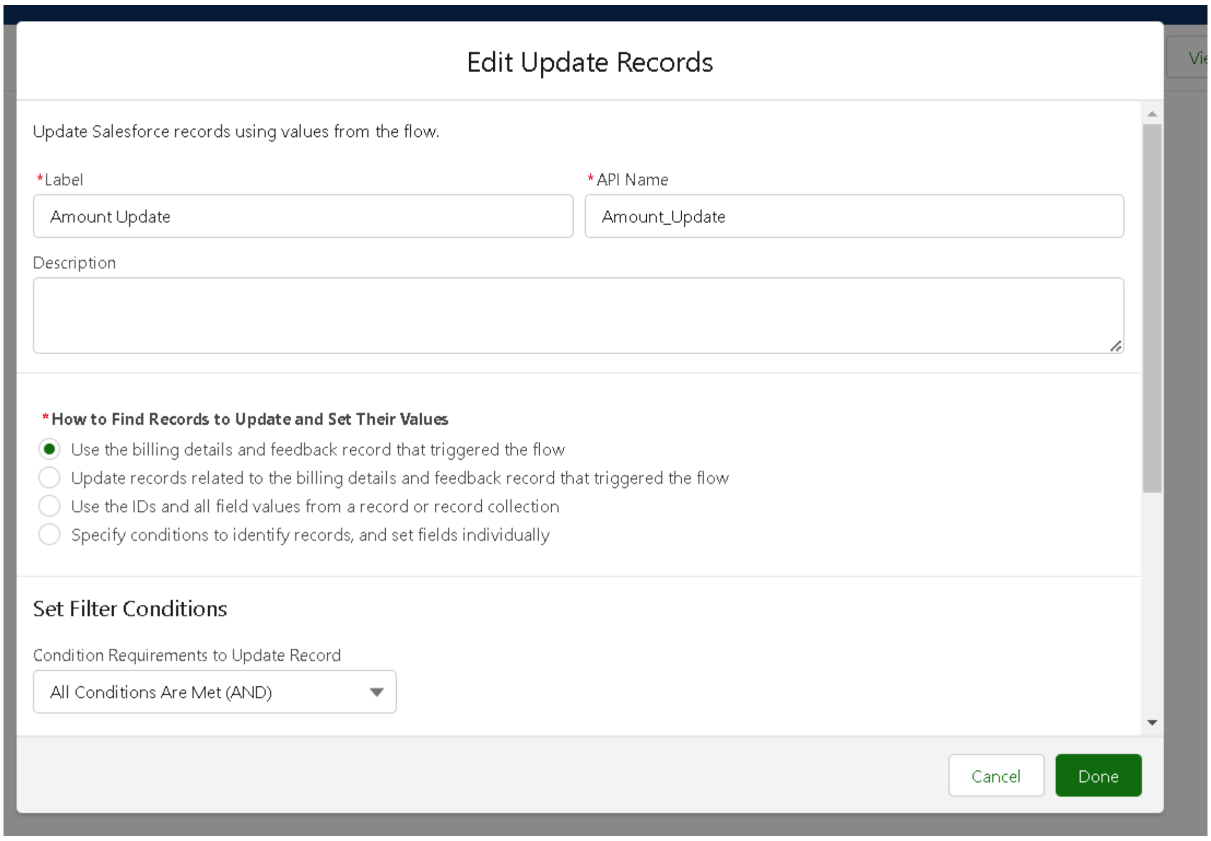
Configure Trigger

*Trigger the Flow When:

- ○ A record is created
- ○ A record is updated
- ● A record is created or updated
- ○ A record is deleted

Under the Record-triggered Flow Click on “+” Symbol and In the Drop down List select the “Update records Element”.

Give the Label Name : Amount Update
Api name : is auto populated



Edit Update Records

Update Salesforce records using values from the flow.

*Label: Amount Update

*API Name: Amount_Update

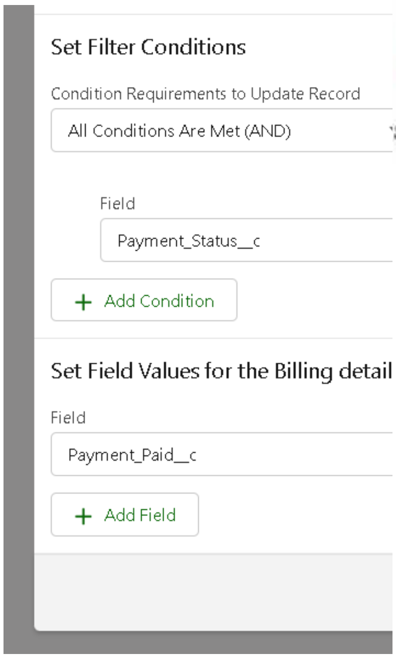
Description:

*How to Find Records to Update and Set Their Values

- [✓ Use the billing details and feedback record that triggered the flow]
- Update records related to the billing details and feedback record that triggered the flow
- Use the IDs and all field values from a record or record collection
- Specify conditions to identify records, and set fields individually

Set Filter Conditions

Condition Requirements to Update Record: All Conditions Are Met (AND)


Set Filter Conditions

Condition Requirements to Update Record

All Conditions Are Met (AND)

Field

Payment_Status__c

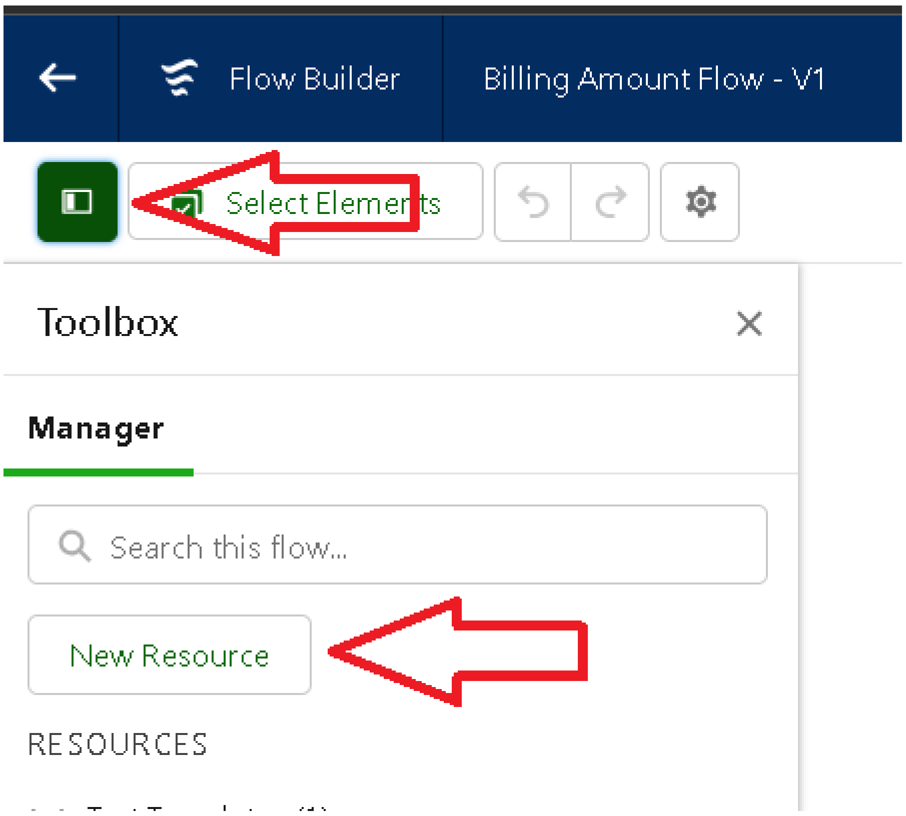
Set Field Values for the Billing detail

Field

Payment_Paid__c

Set a filter condition : All Conditions are met(AND)
Field :Payment_Status__c
Operator : Equals
Value : Completed
And Set Field Values for the Billing details and feedback Record
Field :Payment_Paid__c
Value : {!$Record.Service_records__r.Appointment__r.Service_Amount__c}
Click On Done.

17. Before creating another Element. Create a New Resource form Toolbox form top left.



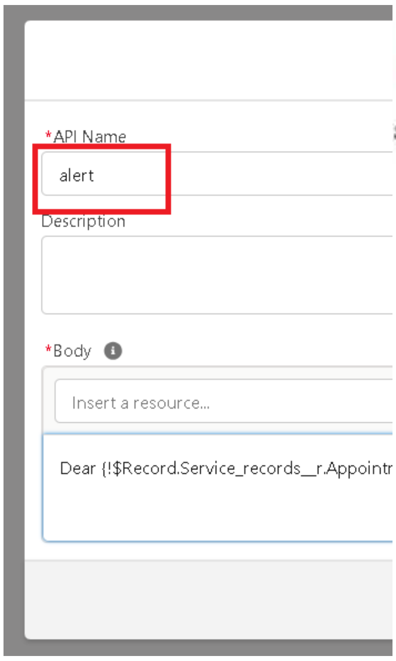
Click on the New Resource, And select Variable.
Select the resource type as text template.
Enter the API name as " alert".
Change the view as Rich Text ? View to Plain Text.
In body field paste the syntax that given below.

Dear {!$Record.Service_records__r.Appointment__r.Customer_Name__r.Name},

I hope this message finds you well. I wanted to take a moment to express my sincere gratitude for your recent payment for the services provided by our garage management team. Your prompt payment isgreatly appreciated, and it helps us continue to provide top-notch services to you and all our valued customers.

Amount paid : {!$Record.Payment_Paid__c}
Thank you for Coming .

Click done.


*API Name

alert

Description

*Body

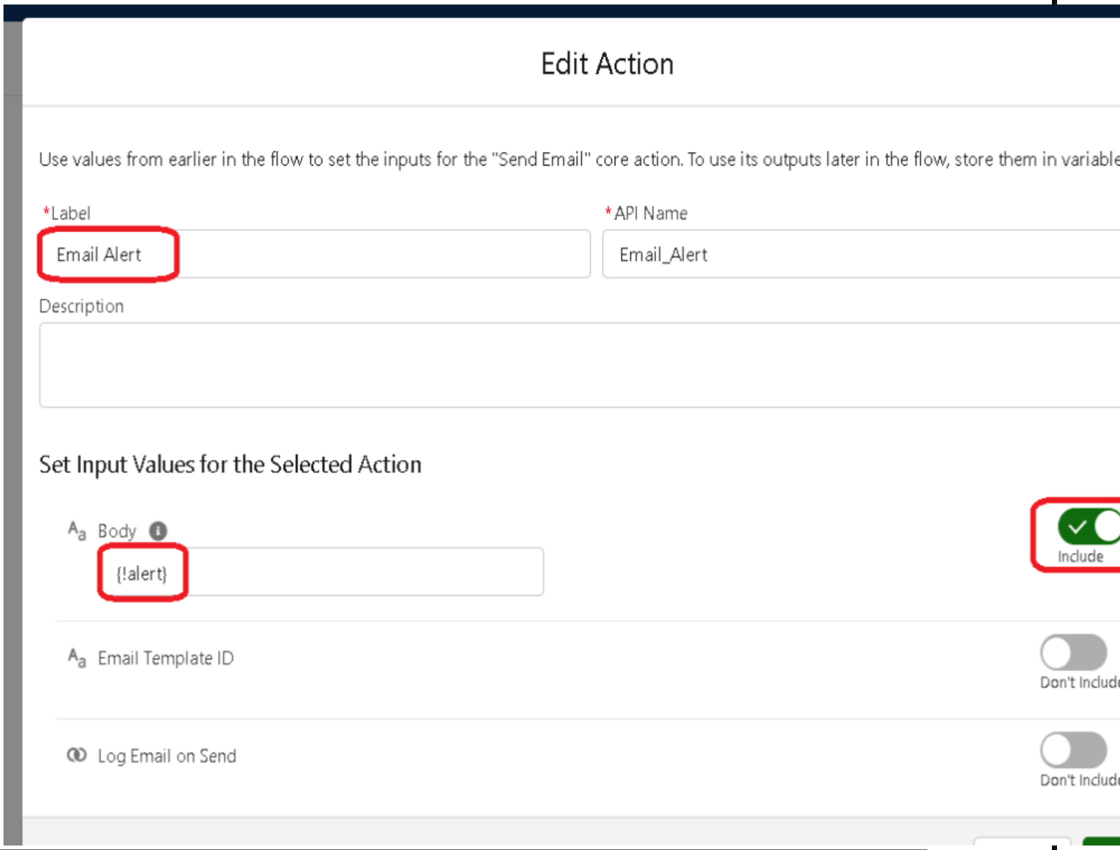
Insert a resource...

Dear {!$Record.Service_records__r.Appointr

Now Click on Add Element , select Action.
Their action bar will be opened in that search for “ send email ” and click on it.
Give the label name as “ Email Alert”
API name will be auto populated.
Enable the body in set input values for the selected action.
Select the text template that created , Body : {!alert}

Include recipient address list select the email form the record.
RecipientAddressList:{!$Record.Service_records__r.Appointment__r.Customer_Name__r.Gmail__c}
Include subject as “ Thank You for Your Payment - Garage Management”.

Click done.


Edit Action

Use values from earlier in the flow to set the inputs for the "Send Email" core action. To use its outputs later in the flow, store them in variable

*Label: Email Alert

*API Name: Email_Alert

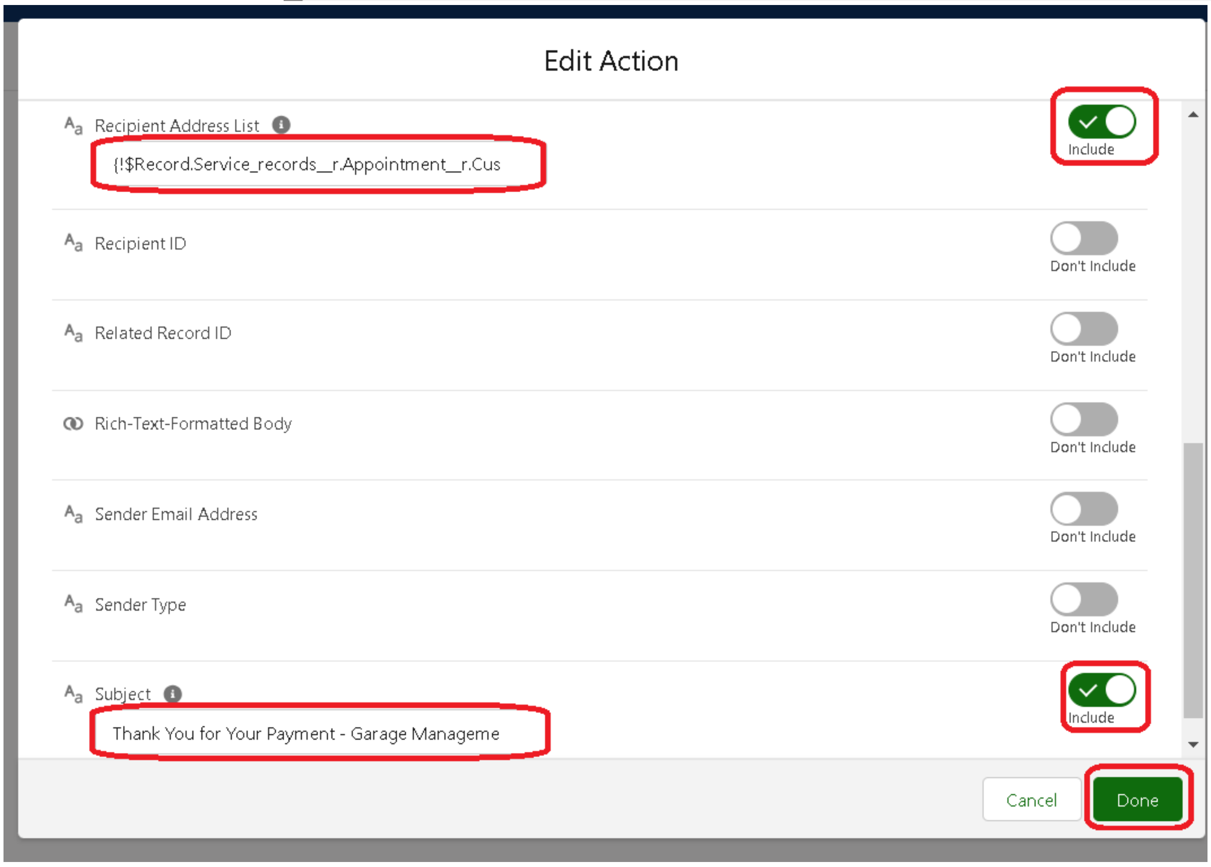
Description

Set Input Values for the Selected Action

Body: {!alert} — Include

Email Template ID — Don't Include

Log Email on Send — Don't Include


Edit Action

Recipient Address List: {!$Record.Service_records__r.Appointment__r.Cus — Include

Recipient ID — Don't Include

Related Record ID — Don't Include

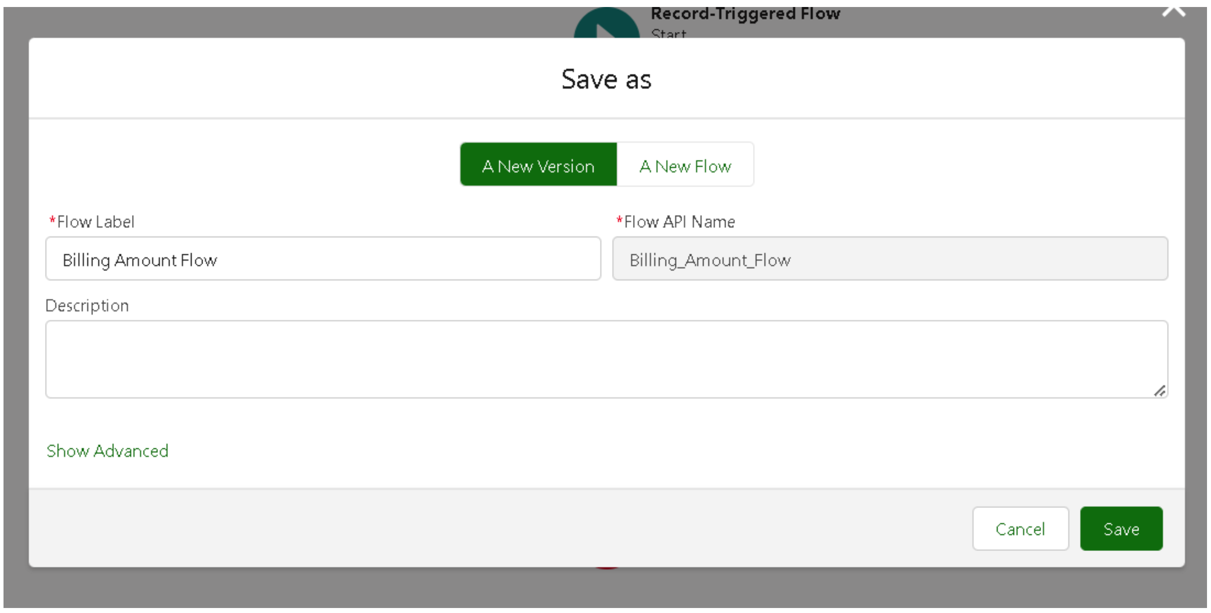
Rich-Text-Formatted Body — Don't Include

Sender Email Address — Don't Include

Sender Type — Don't Include

Subject: Thank You for Your Payment - Garage Manageme — Include

Click on save. Give the Flow label , Flow Api name will be autopopulated.
And click save, and click on activate.



*Flow Label: Billing Amount Flow

*Flow API Name: Billing_Amount_Flow

Description:

Step 9:Apex Trigger

Apex can be invoked by using triggers. Apex triggers enable you to perform custom actions

before or after changes to Salesforce records, such as insertions, updates, or deletions.

A trigger is Apex code that executes before or after the following types of operations:

- insert
- update
- delete

- merge
- upsert
- undelete

For example, you can have a trigger run before an object's records are inserted into the database, after records have been deleted, or even after a record is restored from the Recycle Bin.

You can define triggers for top-level standard objects that support triggers, such as a Contact or an Account, some standard child objects, such as a CaseComment, and custom objects. To define a trigger, from the object management settings for the object whose triggers you want to access, go to Triggers.

There are primarily two types of Apex Triggers:

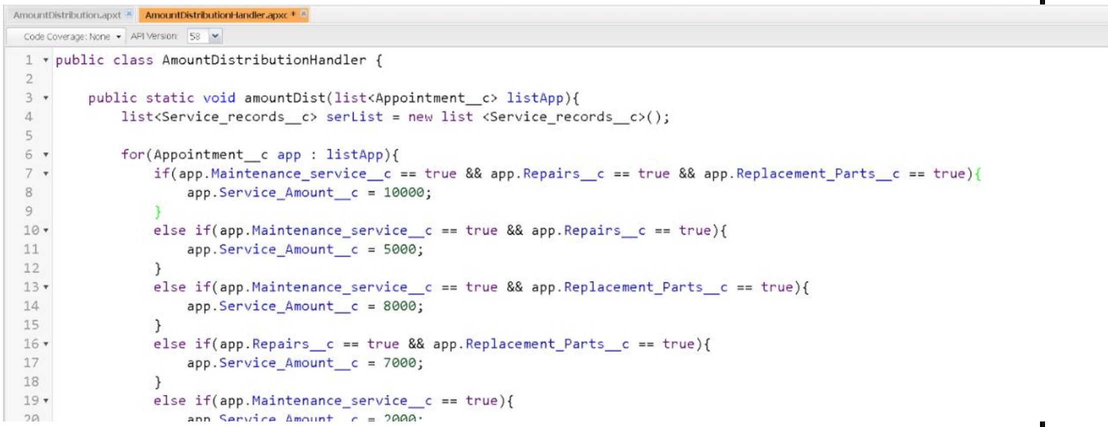
Before Trigger: This type of trigger in Salesforce is used either to update or validate the values of a record before they can be saved into the database. So, basically, the before trigger validates the record first and then saves it. Some criteria or code can be set to check data before it gets ready to be inserted into the database.

After Trigger: This type of trigger in Salesforce is used to access the field values set by the system and affect any change in the record. In other words, the after trigger makes changes to the value from the data inserted in some other record.

Apex Handler

UseCase : This use case works for Amount Distribution for each Service the customer selected for there Vehicle.

1. Login to the respective trailhead account and navigate to the gear icon in the top right corner.
2. Click on the Developer console. Now you will see a new console window.
3. In the toolbar, you can see FILE. Click on it and navigate to new and create New apex class.
4. Name the class as “AmountDistributionHandler ”.



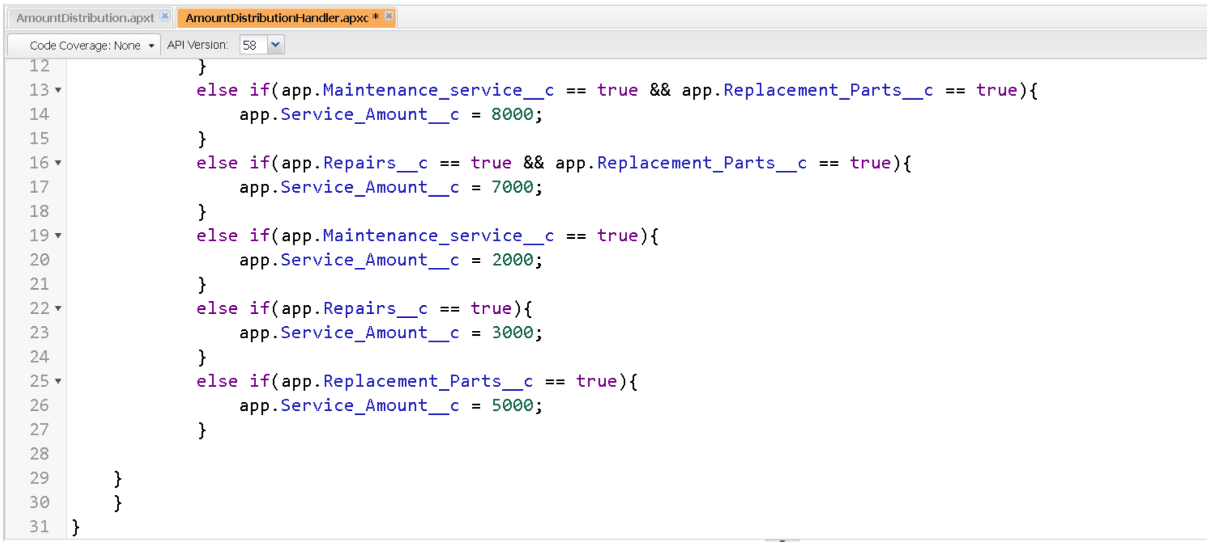
AmountDistribution.apxt | AmountDistributionHandler.apxc *

Code Coverage: None | API Version: 58

```
public class AmountDistributionHandler {

    public static void amountDist(list<Appointment__c> listApp){
        list<Service_records__c> serList = new list <Service_records__c>();

        for(Appointment__c app : listApp){
            if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 10000;
            }
            else if(app.Maintenance_service__c == true && app.Repairs__c == true){
                app.Service_Amount__c = 5000;
            }
            else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 8000;
            }
            else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 7000;
            }
            else if(app.Maintenance_service__c == true){
                app.Service_Amount__c = 2000;
```



AmountDistribution.apxt | AmountDistributionHandler.apxc *

Code Coverage: None | API Version: 58

```
            }
            else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 8000;
            }
            else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 7000;
            }
            else if(app.Maintenance_service__c == true){
                app.Service_Amount__c = 2000;
            }
            else if(app.Repairs__c == true){
                app.Service_Amount__c = 3000;
            }
            else if(app.Replacement_Parts__c == true){
                app.Service_Amount__c = 5000;
            }

        }
    }
}
```

Code:

```
public class AmountDistributionHandler {

 public static void amountDist(list<Appointment__c>listApp){
  list<Service_records__c>serList = new list <Service_records__c>();

  for(Appointment__c app : listApp){
   if(app.Maintenance_service__c == true &&app.Repairs__c == true
&&app.Replacement_Parts__c == true){
    app.Service_Amount__c = 10000;
```

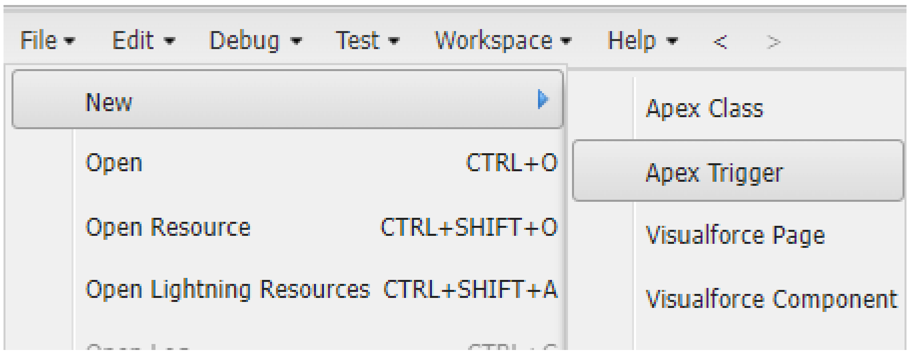
```
                        }
                        else
                  if(app.Maintenance_s
ervice__c == true &&app.Repairs__c == true){
        app.Service_Amount__c = 5000;
      }
      else if(app.Maintenance_service__c == true
&&app.Replacement_Parts__c == true){
        app.Service_Amount__c = 8000;
      }
      else if(app.Repairs__c == true &&app.Replacement_Parts__c ==
true){
        app.Service_Amount__c = 7000;
      }
      else if(app.Maintenance_service__c == true){
        app.Service_Amount__c = 2000;
      }
      else if(app.Repairs__c == true){
        app.Service_Amount__c = 3000;
      }
      else if(app.Replacement_Parts__c == true){
        app.Service_Amount__c = 5000;
      }


  }
  }
}
```

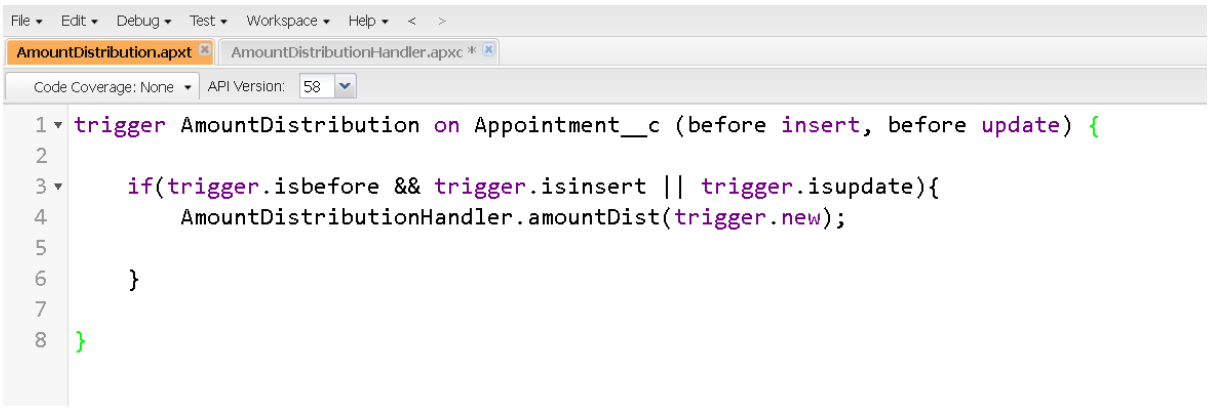
Trigger Handler :

How to create a new trigger :

1. While still in the trailhead account, navigate to the gear icon in the top right corner.
2. Click on developer console and you will be navigated to a new console window.
3. Click on File menu in the tool bar, and click on new? Trigger.
4. Enter the trigger name and the object to be triggered.
5. Name :AmountDistribution
6. sObject :Appointment__c



1. Handler for the Appointment Object



```
trigger AmountDistribution on Appointment__c (before insert, before update) {

    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
        AmountDistributionHandler.amountDist(trigger.new);

    }

}
```

Code:

```
trigger AmountDistribution on
Appointment__c
(before insert, before
update) {

  if(trigger.isbefore&&trigger.isinsert || trigger.isupdate){
    AmountDistributionHandler.amountDist(trigger.new);


  }
```

Reports

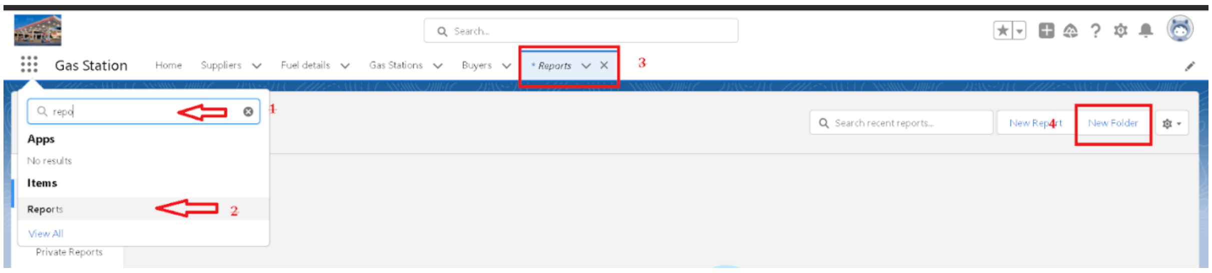
Reports give you access to your Salesforce data. You can examine your Salesforce data in almost infinite combinations, display it in easy-to-understand formats, and share the resulting insights with others. Before building, reading, and sharing reports, review these reporting basics.

Types of Reports in Salesforce

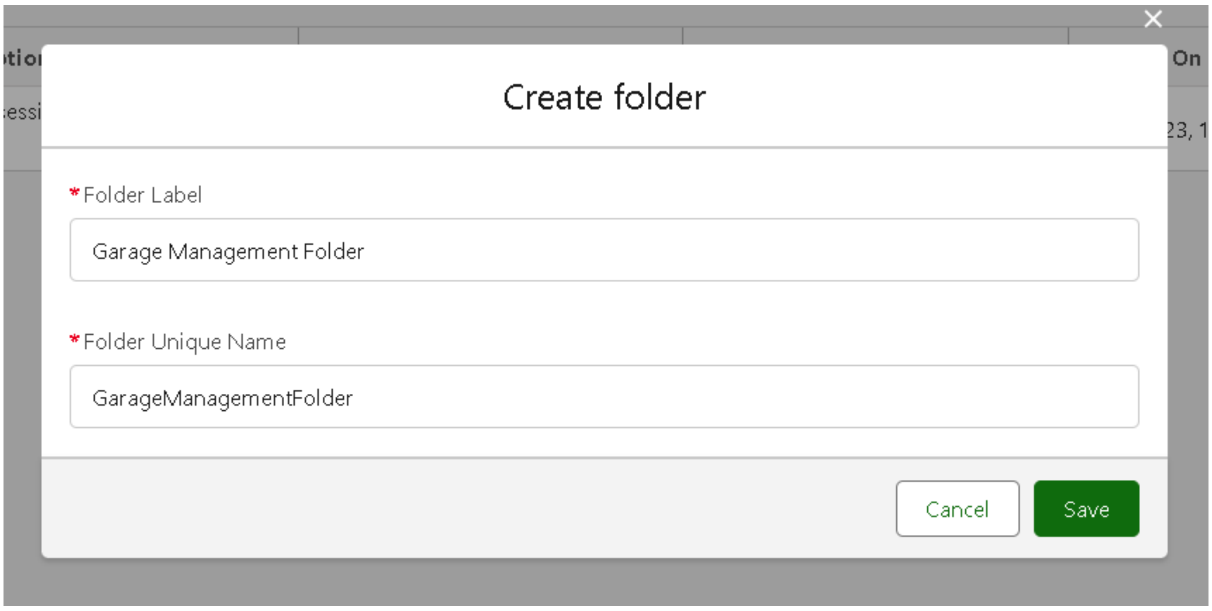
1. Tabular
2. Summary
3. Matrix
4. Joined Reports

create a report folder

1. Click on the app launcher and search for reports.
2. Click on the report tab, click on new folder.



3. Give the Folder label as “Garage Management Folder”, Folder unique name will be auto populated.
4. Click save.



Create folder

*Folder Label

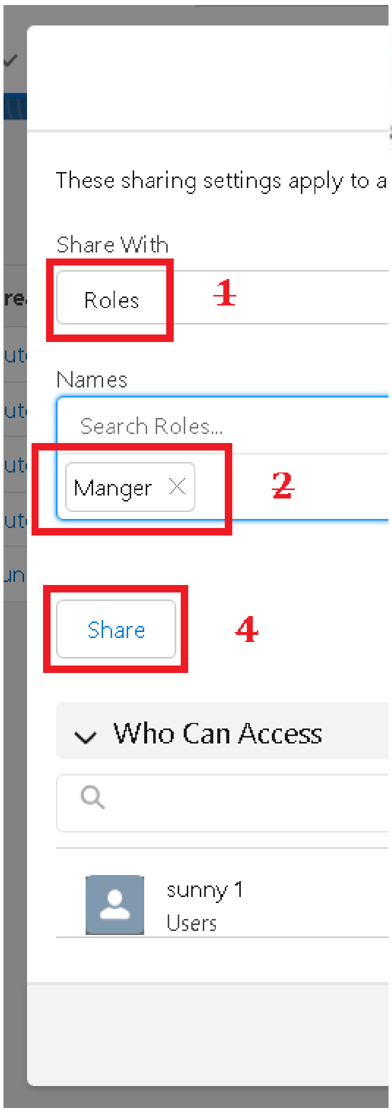
Garage Management Folder

*Folder Unique Name

GarageManagementFolder

Sharing a report folder

1. Go to the app >> click on the reports tab.
2. Click on the All folder , click on the Drop down arrow for Garage Management folder, and Click on share.
3. Select the share with as “roles”, in name field search for “manager”, give “view” as access for that role.
4. Then click share, and click on Done.


These sharing settings apply to a

Share With

Roles 1

Names

Search Roles...

Manger 2

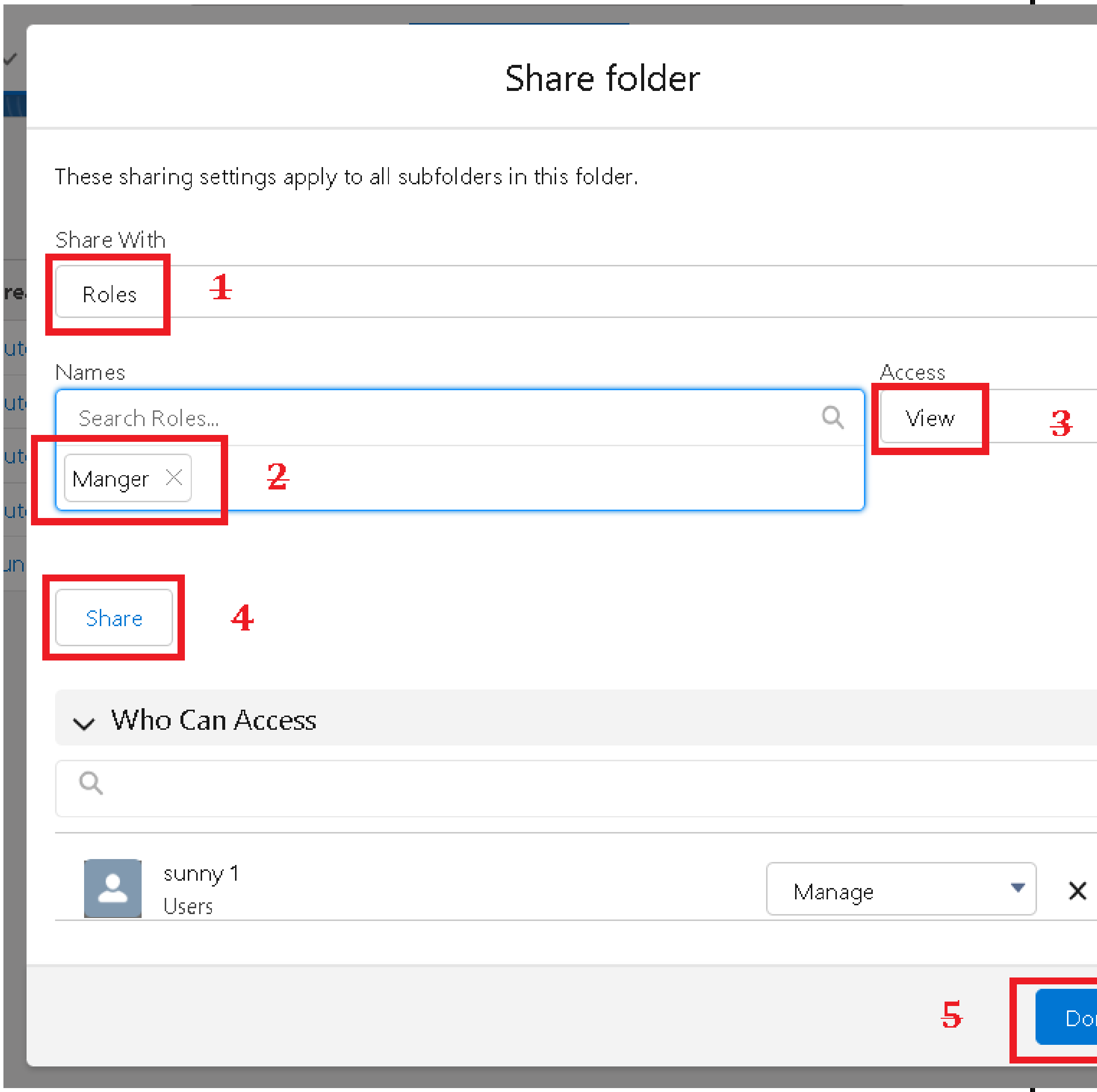
Share 4

Who Can Access

sunny 1
Users

Sharing a report folder

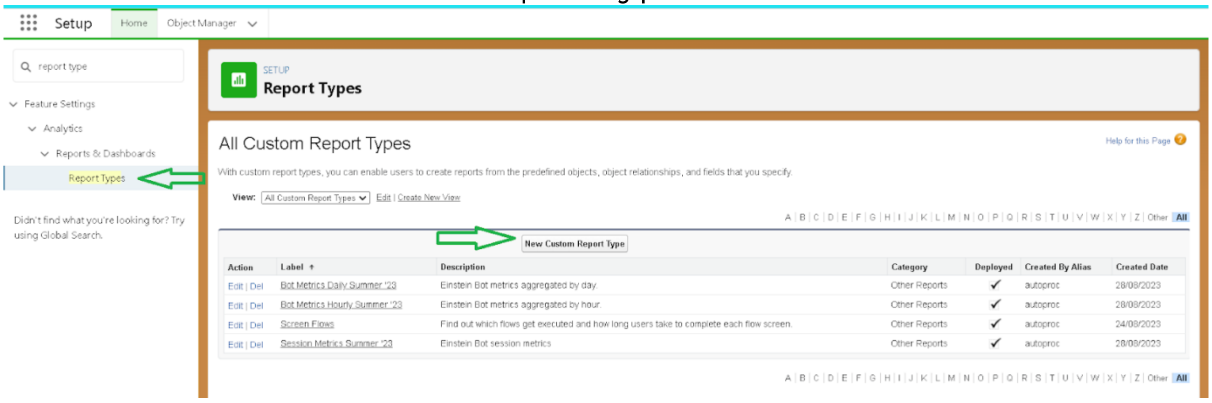
1. Go to the app >> click on the reports tab.
2. Click on the All folder , click on the Drop down arrow for Garage Management folder, and Click on share.
3. Select the share with as “roles”, in name field search for “manager”, give “view” as access for that role.
4. Then click share, and click on Done.



These sharing settings apply to all subfolders in this folder.

Create a report type

1. Go to setup >> type users in quick find box >> select Report Type >> click on Continue.
2. Click on new custom report type.



Didn't find what you're looking for? Try using Global Search.

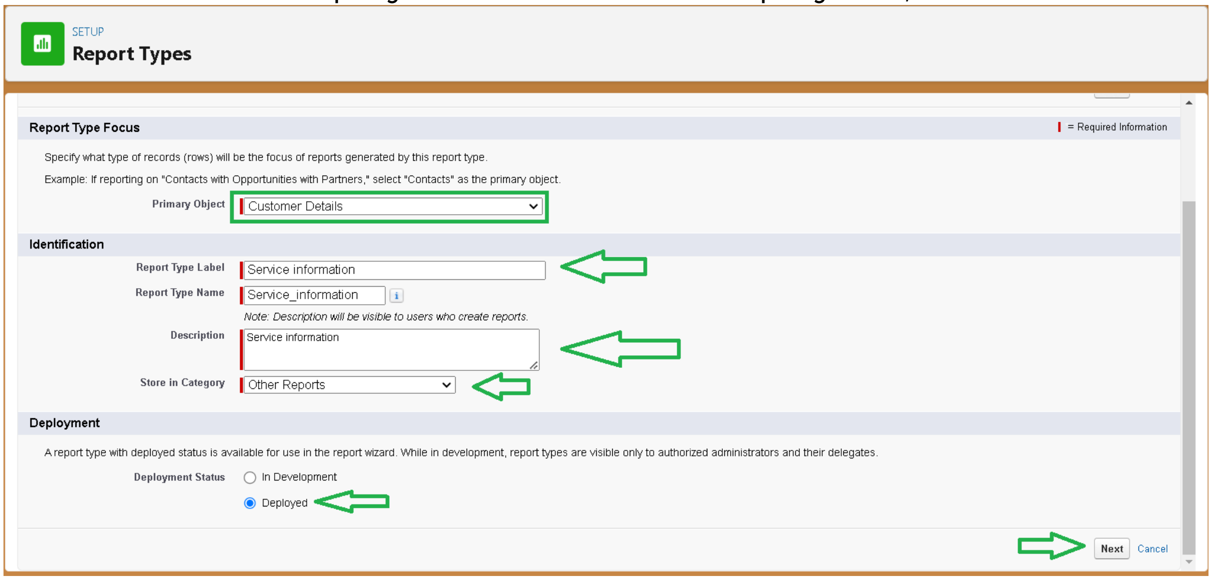
All Custom Report Types

With custom report types, you can enable users to create reports from the predefined objects, object relationships, and fields that you specify.

Action	Label ↑	Description	Category	Deployed	Created By Alias	Created Date
Edit | Del	Bot Metrics Daily Summer '23	Einstein Bot metrics aggregated by day.	Other Reports	✓	autoproc	28/08/2023
Edit | Del	Bot Metrics Hourly Summer '23	Einstein Bot metrics aggregated by hour.	Other Reports	✓	autoproc	28/08/2023
Edit | Del	Screen Flows	Find out which flows get executed and how long users take to complete each flow screen.	Other Reports	✓	autoproc	24/08/2023
Edit | Del	Session Metrics Summer '23	Einstein Bot session metrics	Other Reports	✓	autoproc	28/08/2023

3. Select the Primary object as “ Customer details” .

4. Give the Report type Label as " Service information "
5. Report type Name is autopopulated.
6. Keep the Description as same.
7. Select Store in Category as " other Reports "
8. Select the deployment status as " Depolyed ", click on Next.


SETUP
Report Types

Report Type Focus — = Required Information

Specify what type of records (rows) will be the focus of reports generated by this report type.

Example: If reporting on "Contacts with Opportunities with Partners," select "Contacts" as the primary object.

Primary Object: Customer Details

Identification

Report Type Label: Service information

Report Type Name: Service_information

Note: Description will be visible to users who create reports.

Description: Service information

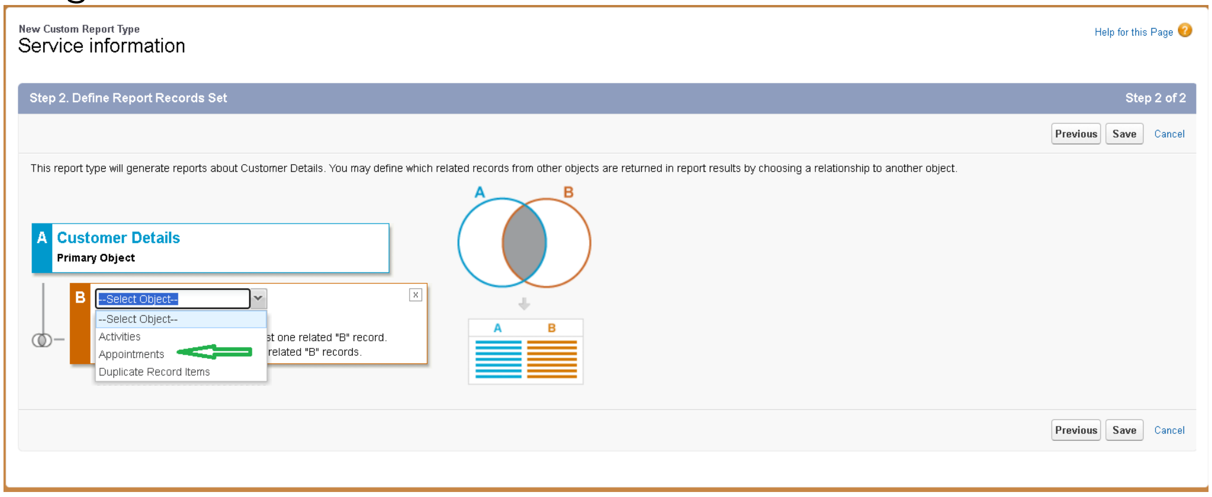
Store in Category: Other Reports

Deployment

A report type with deployed status is available for use in the report wizard. While in development, report types are visible only to authorized administrators and their delegates.

Deployment Status: In Development [✓ Deployed]

9. now , Click on Related object box.
10. Click on Select Object, choose Appointment Object as shown in fig.


New Custom Report Type
Service information

Help for this Page

Step 2. Define Report Records Set — Step 2 of 2

This report type will generate reports about Customer Details. You may define which related records from other objects are returned in report results by choosing a relationship to another object.

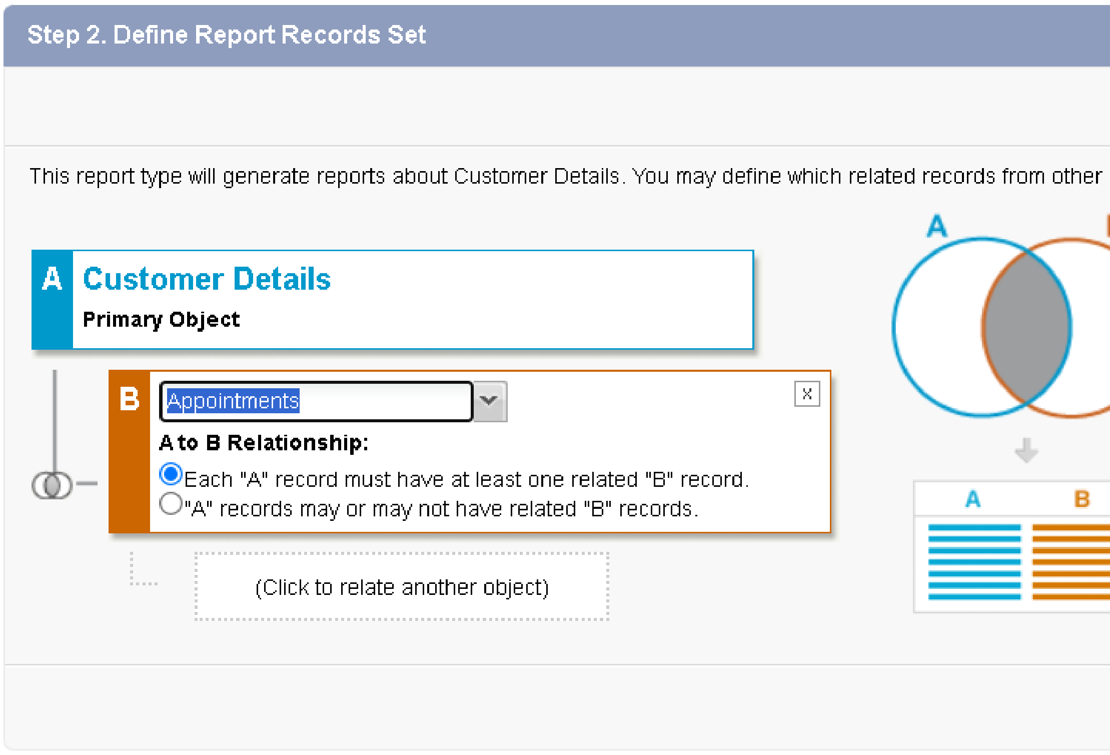
A Customer Details — Primary Object

B --Select Object--

--Select Object--
Activities
Appointments
Duplicate Record Items

...st one related "B" record.
...related "B" records.

Smart Internz


Step 2. Define Report Records Set

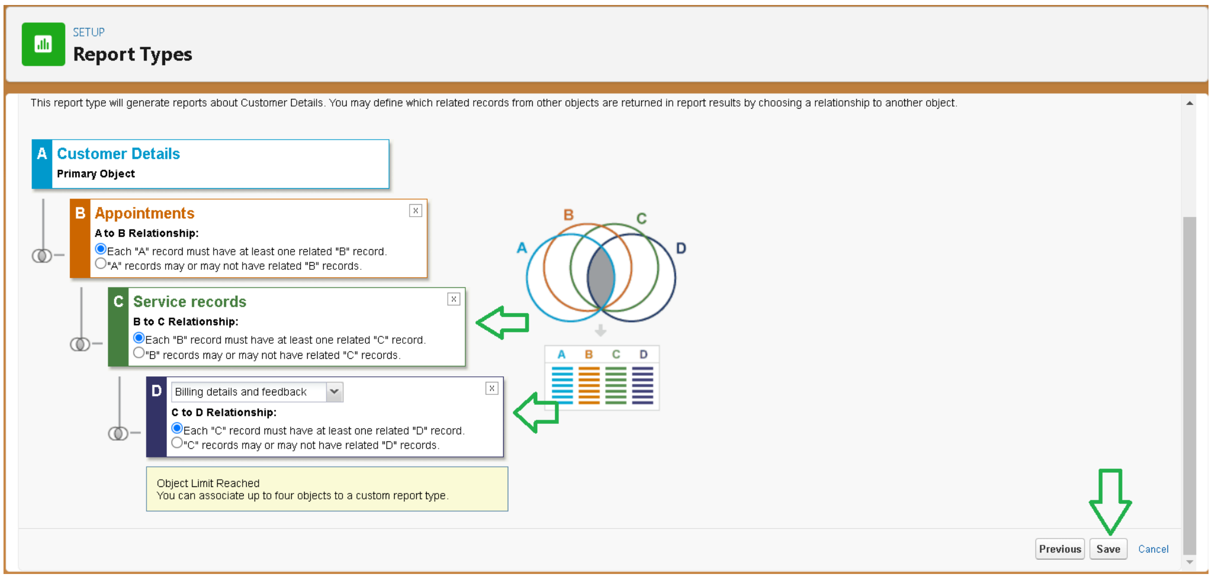
This report type will generate reports about Customer Details. You may define which related records from other

A Customer Details
Primary Object

B Appointments
A to B Relationship:
[✓ Each "A" record must have at least one related "B" record.]
"A" records may or may not have related "B" records.

(Click to relate another object)

11. Again Click to relate another object.
12. And select the related object as “ service records”.
13. Repeat the process and select the related object as “ Billing details and feedback”.
14. And click on save.


SETUP
Report Types

This report type will generate reports about Customer Details. You may define which related records from other objects are returned in report results by choosing a relationship to another object.

A Customer Details
Primary Object

B Appointments
A to B Relationship:
[✓ Each "A" record must have at least one related "B" record.]
"A" records may or may not have related "B" records.

C Service records
B to C Relationship:
[✓ Each "B" record must have at least one related "C" record.]
"B" records may or may not have related "C" records.

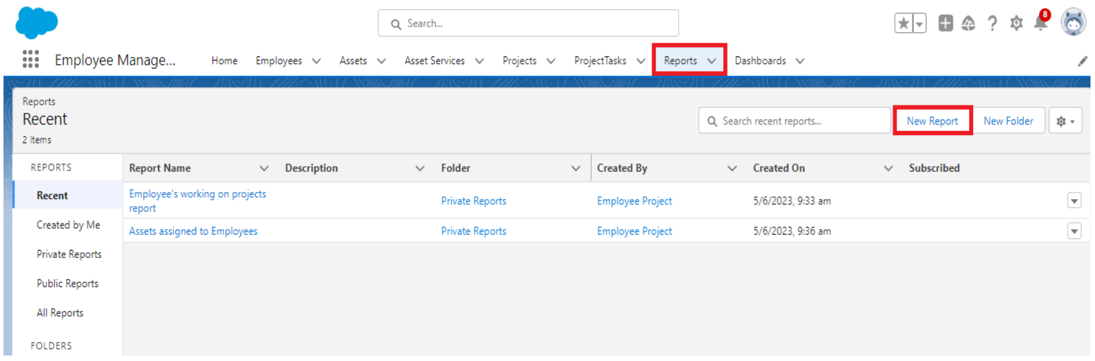
D Billing details and feedback
C to D Relationship:
[✓ Each "C" record must have at least one related "D" record.]
"C" records may or may not have related "D" records.

Object Limit Reached
You can associate up to four objects to a custom report type.

Create report

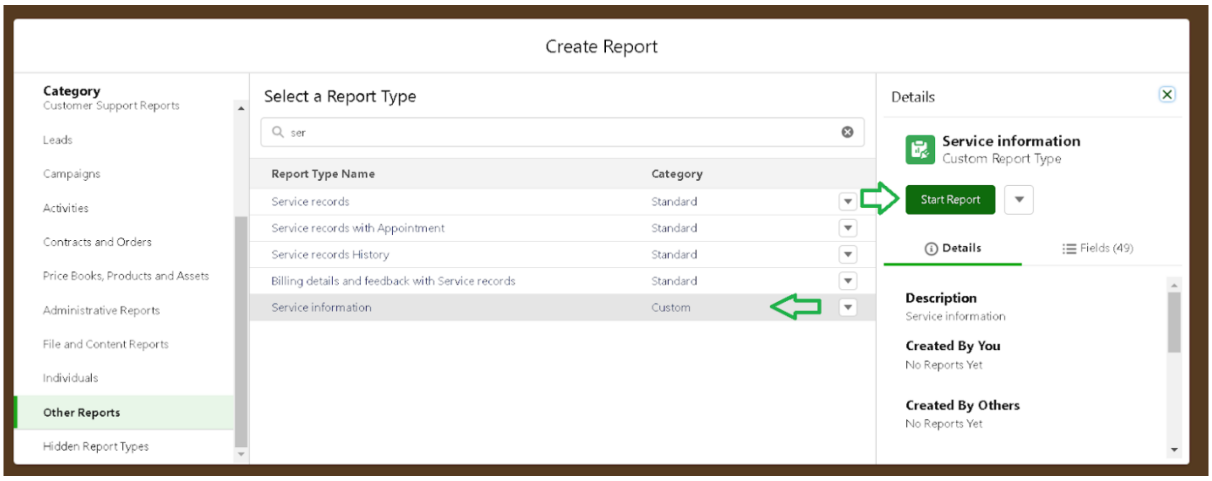
Note : Before creating report, create latest "10" records in every object.
Try to fill every field in each record for better experience.

1. Go to the app >> click on the reports tab
2. Click New Report.



Report Name	Description	Folder	Created By	Created On	Subscribed
Employee's working on projects report		Private Reports	Employee Project	5/6/2023, 9:33 am	
Assets assigned to Employees		Private Reports	Employee Project	5/6/2023, 9:36 am	

3. Select the Category as other reports, search for Service Information, select that report, click on it. And click on start report.



Report Type Name	Category
Service records	Standard
Service records with Appointment	Standard
Service records History	Standard
Billing details and feedback with Service records	Standard
Service information	Custom

Description
Service information
Created By You
No Reports Yet
Created By Others
No Reports Yet

4. Their outline pane is opened alredy, select the fields that mentioned below in column section.

a. Customer name

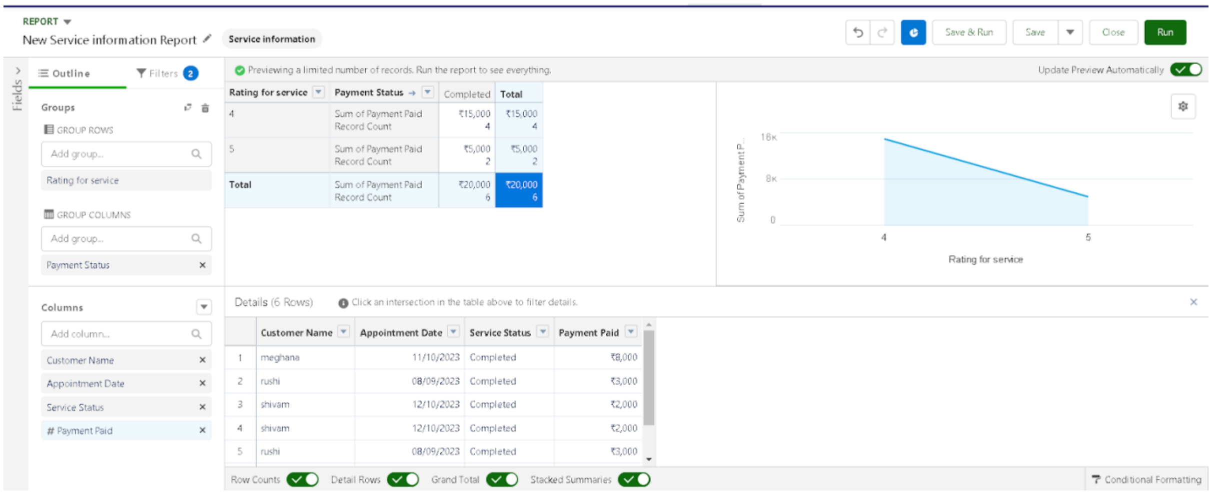
b. Appointment Date

c. Service Status

d. Payment paid

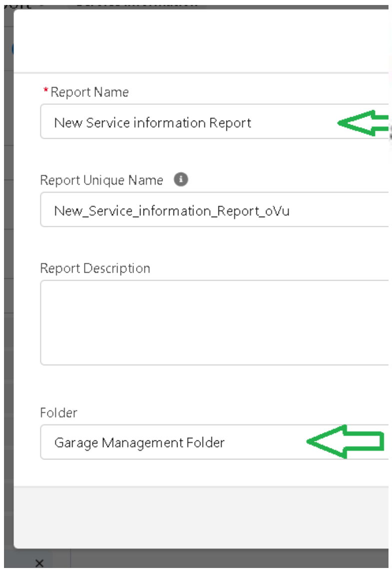
5. Remove the unnecessary fields.
6. Select the fields that mentioned below in GROUP ROWS section.
   . Rating for Service
7. Select the fields that mentioned below in GROUP ROWS section.
   . Payment Status
8. Click on Add Chart , Select the Line Chart.

9. Click on save, Give the report Name : New Service information Report
10. Report unique Name is auto populated.
11. Select the folder the created and Click on save.



Rating for service	Payment Status →	Completed	Total
4	Sum of Payment Paid Record Count	₹15,000 4	₹15,000 4
5	Sum of Payment Paid Record Count	₹5,000 2	₹5,000 2
Total	Sum of Payment Paid Record Count	₹20,000 6	₹20,000 6

	Customer Name	Appointment Date	Service Status	Payment Paid
1	meghana	11/10/2023	Completed	₹8,000
2	rushi	08/09/2023	Completed	₹3,000
3	shivam	12/10/2023	Completed	₹2,000
4	shivam	12/10/2023	Completed	₹2,000
5	rushi	08/09/2023	Completed	₹3,000


*Report Name
New Service information Report
Report Unique Name
New_Service_information_Report_oVu
Report Description
Folder
Garage Management Folder

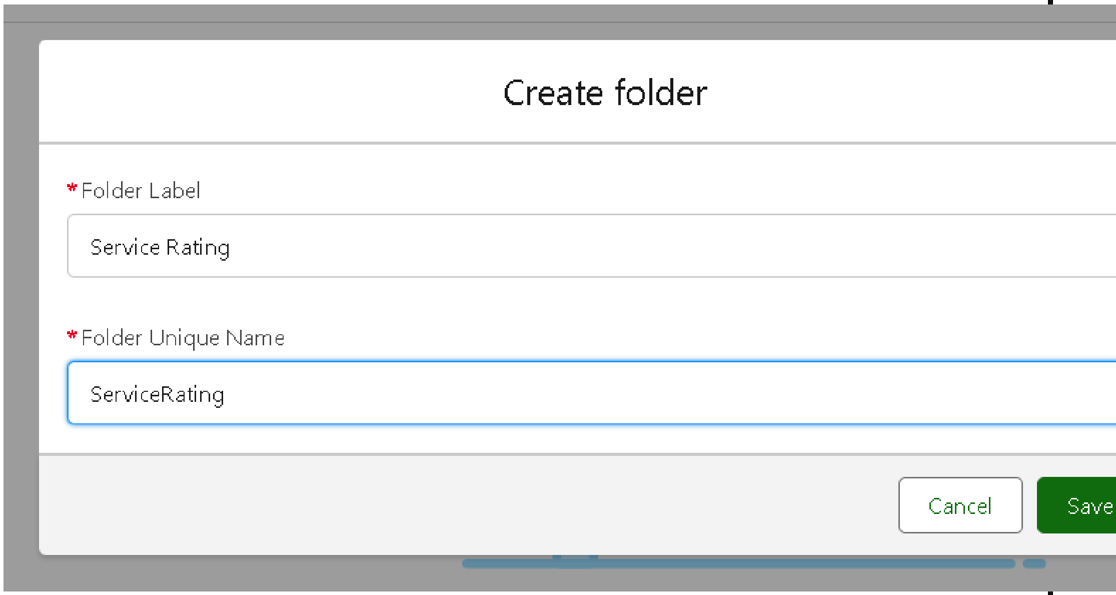
Dashboards

Dashboards help you visually understand changing business conditions so you can make decisions based on the real-time data you've gathered with reports. Use dashboards to help users identify trends, sort out quantities, and measure the impact of their activities. Before building, reading, and sharing dashboards, review these dashboard basics.

Create a dashboard folder

1. Click on the app launcher and search for dashboard.
2. Click on dashboard tab.
3. Click new folder, give the folder label as " Service Rating dashboard".
4. Folder unique name will be auto populated.
5. Click save.

Smart Internz


Create folder

* Folder Label

Service Rating

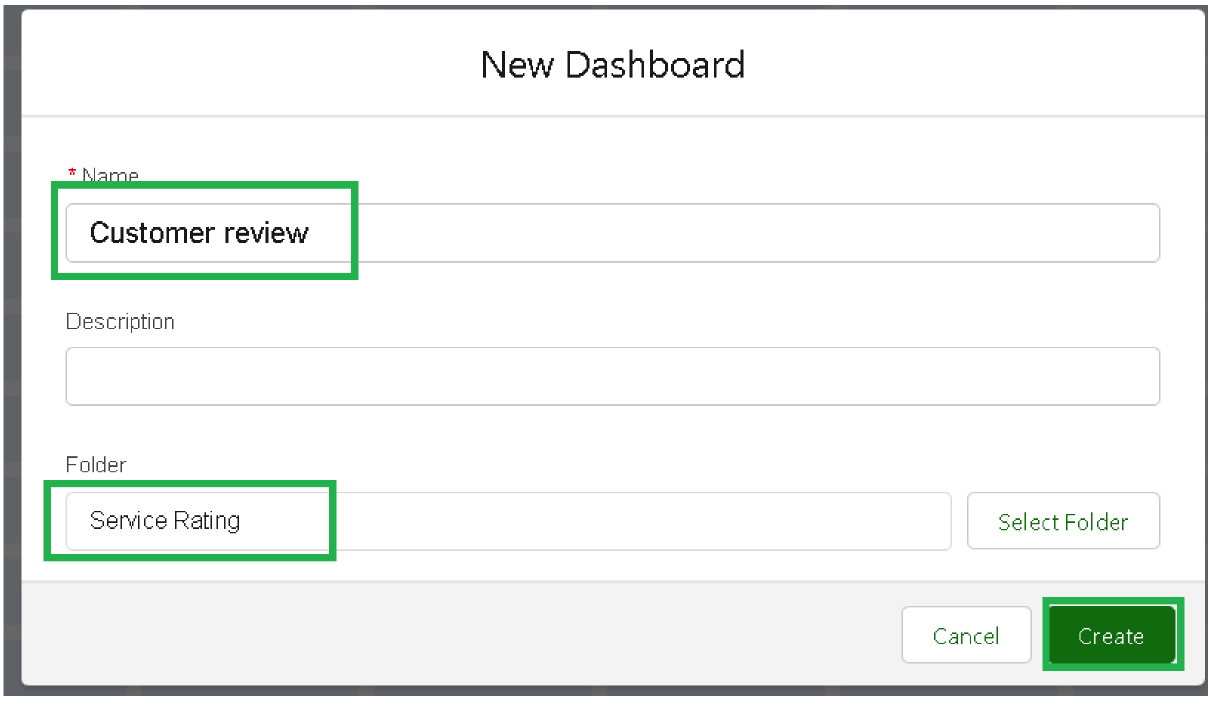
* Folder Unique Name

ServiceRating

6. Follow the same steps, form milestone 15, and activity 2, and provide the sharing settings for the folder that just created.

Creation Dashboard

1. Go to the app >> click on the Dashboards tabs.
2. Give a Name and select the folder that created, and click on create.


New Dashboard

* Name

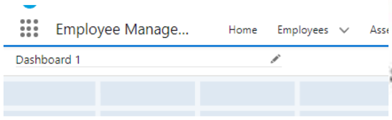
Customer review

Description

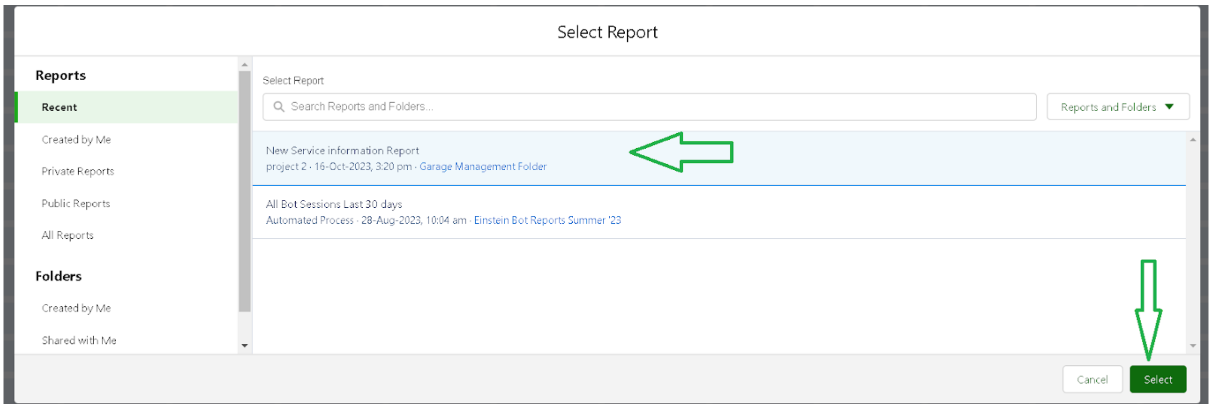
Folder

Service Rating

3. Select add component.



4. Select a Report and click on select.



Select Report

Reports
Recent
Created by Me
Private Reports
Public Reports
All Reports
Folders
Created by Me
Shared with Me

Select Report

Search Reports and Folders...

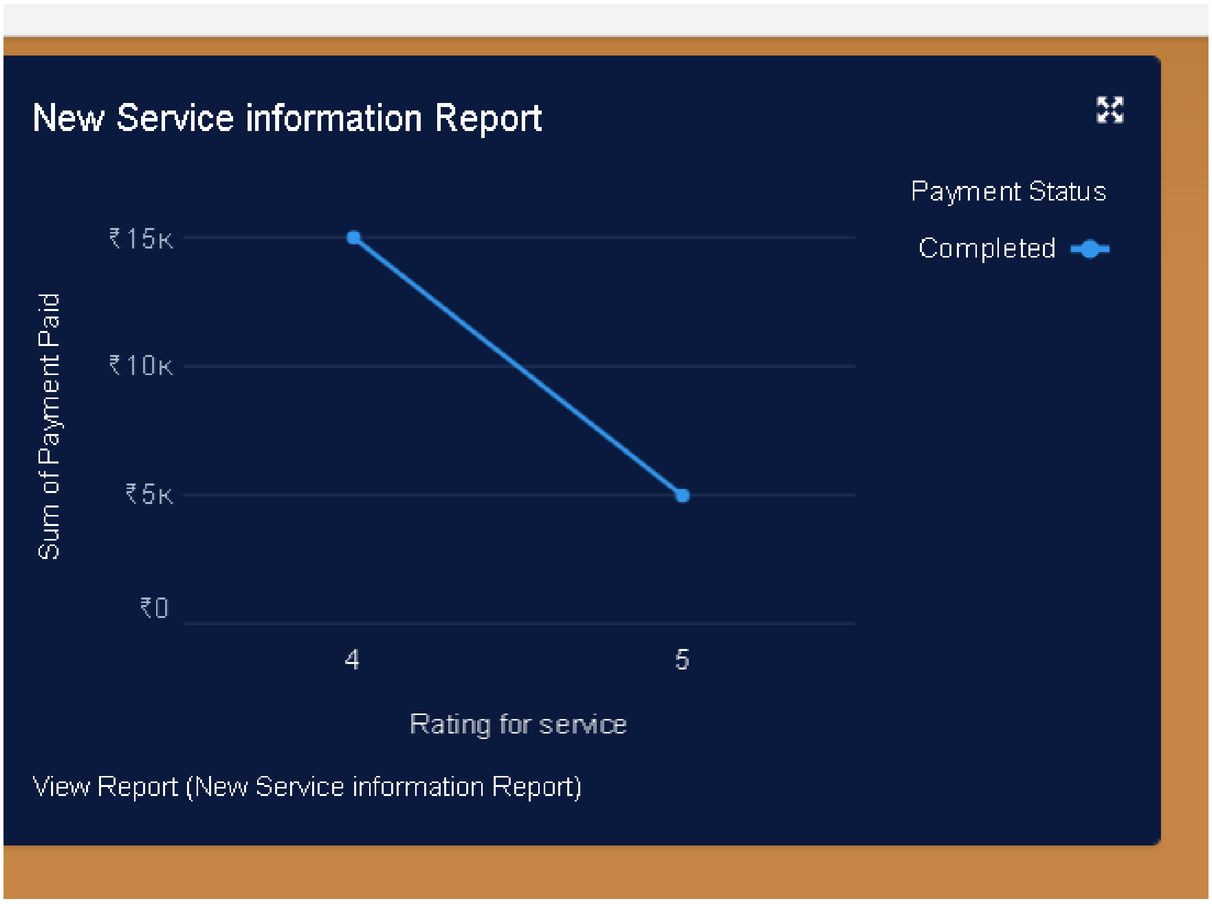
Reports and Folders

New Service information Report
project 2 · 16-Oct-2023, 3:20 pm · Garage Management Folder

All Bot Sessions Last 30 days
Automated Process · 28-Aug-2023, 10:04 am · Einstein Bot Reports Summer '23

Cancel Select

5. Select the Line Chart. Change the theme.
6. Click Add then click on Save and then click on Done.
7. Preview is shown below.

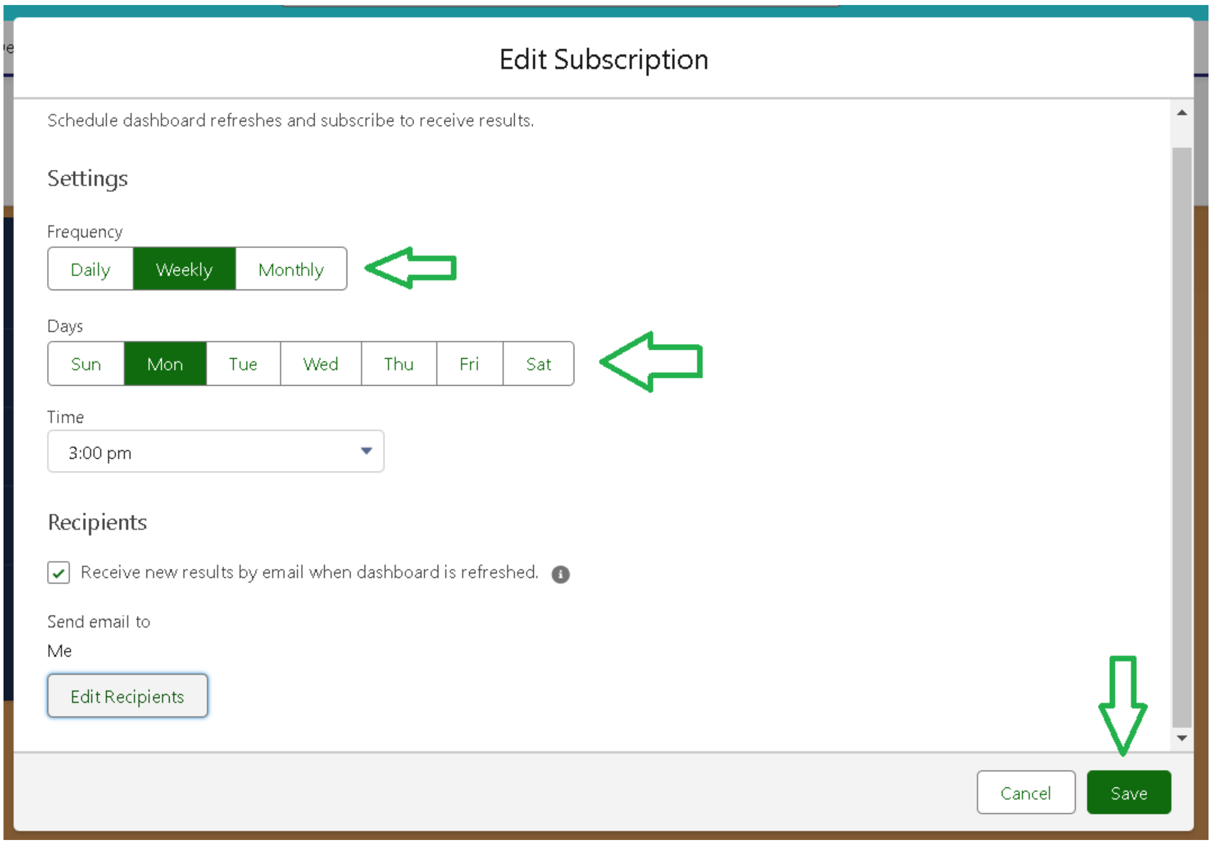


8. After that Click on Subcribe on top right.

9. Set the Frequency as " weekly ".

10. Set a day as monday.

11. And Click on save.



Edit Subscription

Schedule dashboard refreshes and subscribe to receive results.

Settings

Frequency: Daily [✓ Weekly] Monthly

Days: Sun [✓ Mon] Tue Wed Thu Fri Sat

Time: 3:00 pm

Recipients

✓ Receive new results by email when dashboard is refreshed.

Send email to

Me